НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Інститут телекомунікаційних систем

Кафедра Телекомунікаційних систем

«На правах рукопису»
УДК 621.319

«До захисту допущено»
Завідувач кафедри
_____ Л.О. Уривський
«___»_____20__ р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 172 Телекомунікації та радіотехніка

на тему: «Прикладні аспекти реалізації рішень передавання інформації в технологіях Інтернету речей»

Виконав (-ла):
студент (-ка) ІІ курсу, групи ТС-81мп
Киращук Василь Васильович _____

Керівник:
Доцент кафедри ТС, к.т.н., доцент
Осипчук С.О.
_____

Рецензент:
Доцент кафедри ТК, к.т.н., доцент
Міночкін Д.А. _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2019 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут телекомунікаційних систем
Кафедра Телекомунікаційних систем

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність (спеціалізація) – 172 «Телекомунікації та радіотехніка» (172.3620.1 «Телекомунікаційні системи та мережі»)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Л.О. Уривський
«___»_____20__ р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Киращуку Василю Васильовичу

1. Тема дисертації «Прикладні аспекти реалізації рішень передавання інформації в технологіях Інтернету речей», науковий керівник дисертації Осипчук Сергій Олександрович, кандидат технічних наук, доцент кафедри телекомунікаційних систем, затверджені наказом по університету від «___»_____ 20__ р. №_____

2. Термін подання студентом дисертації _____

3. Об’єкт дослідження: технології передавання даних для застосунків Інтернету речей

4. Предмет дослідження: прикладні аспекти реалізації рішень і способів передавання інформації в технологіях Інтернету речей

5. Перелік завдань, які потрібно розробити:

1. розробка комунікаційного модуля для зарядних станцій електромобілів, з метою телеметричного та інших видів контролю і управління станцією.

2. Організації зв’язку з центром керування станцією і сервером обробки даних отриманих від станції.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу

Плакат № 1 «Тема, мета, актуальність, об’єкт, предмет, проблематика, завдання дослідження»»

Плакат № 2 «Актуальність IOT»

Плакат № 3 «Дослідження особливостей організації архітектур та рішень Інтернету речей (1/2)»

Плакат № 4 «Дослідження особливостей організації архітектур та рішень Інтернету речей (2/2)»

Плакат № 5 «Огляд протоколу OCPP»

Плакат № 6 «Архітектура системи моніторингу та управління зарядних станцій»

Плакат № 7 «Висновки»

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Огляд сучасних технологій передавання інформації	01.09.18-31.12.18	
2	Аналіз технології передавання інформації для застосунків інтернету речей	01.10.18-31.12.18	
3	Дослідження особливостей організації архітектур та рішень інтернету речей	01.11.18-31.12.18	
4	Розробка комунікаційного модуля для зарядних станцій та системи контролю для зарядних станцій	01.11.18-31.03.19	
5	Оформлення пояснювальної записки	01.07.19-31.11.19	

Студент Киращук В.В.

Науковий керівник дисертації Осипчук С.О.

РЕФЕРАТ

Текстова частина дипломної роботи: 85 с., 39 рис.., 3 табл., 13 джерел.

Об’єкт дослідження – технології передавання даних для застосунків Інтернету речей

Предмет дослідження – прикладні аспекти реалізації рішень і способів передавання інформації в технологіях Інтернету речей

Мета дослідження – формування рекомендацій щодо вибору технології передавання даних із врахуванням прикладних аспектів реалізації застосунків Інтернету речей

Прикладне завдання - розробка комунікаційного модуля для зарядних станцій електромобілів, з метою телеметричного та інших видів контролю і управління станцією, та організації зв’язку з центром керування станцією і сервером обробки даних отриманих від станції.

Ключові слова**:** IEEE 802.11, ESP8266, WI-FI, OCPP, CHARGE POINT, JSON, IOT, CLOUD, BACKEND.

ABSTRACT

The object of study - data transmission technologies for Internet of Things applications

The subject of the research is applied aspects of realization of decisions and methods of information transfer in the Internet of things technologies

The purpose of the study is to formulate recommendations for the choice of data technology, taking into account the applied aspects of the implementation of the Internet of Things

An application task is to develop a communication module for electric vehicle charging stations for the purpose of telemetry and other types of station monitoring and control, and to establish communication with the station control center and the data server received from the station.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

AI	Artificial intelligence - штучний інтелект
GSM	Global System for Mobile — глобальна система мобільного зв'язку
IEEE	Institute of Electrical and Electronic Engineers - міжнародна організація, що займається розробкою стандартів у галузі електронних технологій
IoT	Internet of things - інтернет речей
LLC	Logical Link Control - управління логічним зв'язком
MAC	Media Access Control - управління доступом до носія
FFD	Повністю функціональні пристрої
RFD	пристрої з обмеженим набором функції
SaaS	software as a service - програмне забезпечення як послуга
PAN	Private Area Network – персональна мережа
LAN	Local Area Network — локальна мережа
MAN	Metropolitan Area Network — мережа в масштабах міста або населеного пункту
ML	Machine learning - Машинне навчання
OCPP	Open Charge Point Protocol - протокол відкритою зарядної точки
WAN	Wide Area Network – глобальна мережа
WiMAX	Worldwide Interoperability for Microwave Access — стандарт безпровідної мережі

ВСТУП

Важко знайти в наш час людину, нічого не чула про Internet of Things (IoT, Інтернет речей). У світових засобах масової інформації говорять про майбутню технологічну революцію, яка торкнеться всіх і змінить наше життя в тій же мірі, як поява телебачення, персональних комп'ютерів або мережі Інтернет. Згідно з найбільш поширеним формулюванням, Інтернет речей (англ. Internet of Things, IoT) - концепція обчислювальної мережі фізичних предметів («речей»), оснащених вбудованими технологіями для взаємодії один з одним або з зовнішнім середовищем, яка розглядає організацію таких мереж як явище, здатне перебудувати економічні та суспільні процеси, що виключає з частини дій і операцій необхідність участі людини. Концепція сформульована в 1999 році як осмислення перспектив широкого застосування засобів радіочастотної ідентифікації для взаємодії фізичних предметів між собою і з зовнішнім оточенням.

Наповнення концепції «інтернету речей» різноманітним технологічним змістом і впровадження практичних рішень для її реалізації починаючи з 2010-х років вважається стійкою тенденцією в інформаційних технологіях, перш за все, завдяки повсюдного поширення бездротових мереж, появи хмарних обчислень, розвитку технологій міжмашинної взаємодії (Machine to machine (M2M)), початку активного переходу на IPv6 і освоєння програмноконфігуруємих мереж. Концепція передбачає, що інтернет речей здатний серйозно вплинути на розвиток сучасного суспільства, оскільки дозволить багатьом процесам відбуватися без участі людини.

РОЗДІЛ 1 ОГЛЯД СУЧАСНИХ ТЕХНОЛОГІЙ ПЕРЕДАВАННЯ ІНФОРМАЦІЇ

1.1 Тенденції розвитку телекоммуникаційних мереж

Як добре відомо, основний недолік аналогових мереж є низькою здатність каналів передачі даних, яка була головною умовою для переходу до цифрових систем зв'язку, що також стимулюється інтенсивним розвитком цифрових технологій.

У зв’язку з тим, що розповсюдження аналогового зв’язку стало значним, був знайдений вихід у розробці дворежимної аналого-цифрової системи, що поєднувала роботу аналогової і цифрової систем в одному діапазоні.

У Європі в кінці 80-х на початку 90-х років минулого століття був розроблений стандарт GSM з використанням діапазону 1800 МГц, який і став в Європі стандартом мобільного зв’язку другого покоління (2G). Зазначений стандарт забезпечив покращення якості звуку, збільшив захищеність мереж зв’язку та підвищив їх продуктивність.

Технічні рішення дозволили збільшити робочу смугу частот, що у поєднанні з меншими розмірами забезпечило побудову стільникових мереж значно більшої ємності. Швидкість передачі даних зросла до 9,6…14,4 Кбіт/с.

Мобільний зв'язок третього покоління (3G) побудовано на основі пакетної передачі даних за стандартом IMT-2000.

Сучасність характеризується динамічною зміною середовища обміну інформацією, тому передбачити, якими будуть у майбутньому телекомунікаційні мережі неможливо, але завдяки тенденціям розвитку технологій та прогнозам майбутніх потреб суспільства можна зробити деякі висновки.

Майбутнє телекомунікаційних мереж

Розвиток мобільних мереж та здешевлення обладнання і послуг для них, призвело до тотальної телефонізації суспільства. Так, відповідно до даних компаній, які надають послуги мобільного зв’язку у 97 країнах світу кількість мобільних пристроїв перевищила чисельність населення. Відповідно до

результатів досліджень Cisco за період з 2011 по 2016 рік обсяг світового мобільного трафіку зріс у 18 разів до 10,8 ексабайт на місяць.

Із поширенням смартфонів та інших мобільних пристроїв люди частіше використовують мобільні пристрої не тільки для відвідування Інтернет-сторінок, але і для перегляду онлайн-відео, використання популярних ресурсів таких, як You Tube, та інших. Це призводить до перенавантаження мережі мобільних операторів, що негативно впливає на якість послуг зв'язку. Водночас, за оцінкою фахівців Alcatel-Lucent, в мережах майбутнього буде в 10…15 разів більше кінцевих пристроїв, що не є мобільними терміналами (телефонами, смартфонами, планшетами). І питання розвитку мереж майбутнього буде залежати саме від їхньої здатності працювати з такими різнорідними компонентами.

Дослідження показали, що користувачі мобільного Інтернету більшу частину трафіку використовували на перегляд онлайн-відео (Рисунок 1.1).

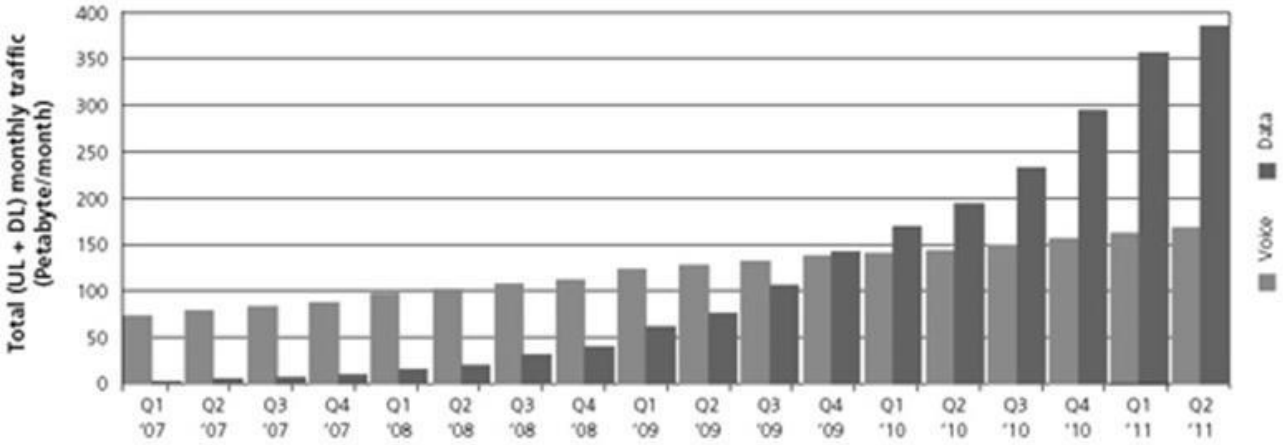

Рисунок 1.1 Аналіз зростання трафіку

На графіку видно, що зростання голосового трафіку значно відстає від зростання трафіку обміну пакетними даними. Ці фактори зумовлюють швидке зростання споживання трафіку абонентами операторів мобільного зв’язку.

Додаткові послуги (VAS-послуги) операторів мобільного зв’язку також здійснюють значне навантаження на мережу. За даними AC&M Comsulting, в Європі за минулий рік обсяг ринку додаткових послуг виріс на 10%, в США - на 25%. Що складає майже четверту частину прибутку операторів. Дослідники

CiscoSystems дійшли висновку, що враховуючи тенденції росту популярності різних додатків, в наступні кілька років потреба в смузі пропускання буде зростати на 300…500% щорічно.

Але не всі технології сьогодення знайдуть місце в мережах завтрашнього дня. Це стосується в першу чергу WiMAX. Підтвердженням цього є обладнання мобільних пристроїв, які сьогодні поступають на ринок. Дуже незначна доля їх обладнана WiMAX і тенденції до збільшення таких пристроїв не спостерігається. А гаджети з підтримкою Wi-Fi практично оволоділи ринком мобільних пристроїв.

Тому можна зробити висновок, що технологія Wi-Fi буде актуальною і в майбутньому, але за умови скоординованої і взаємозалежної роботи безпроводових та стільникових мережі. Це стосується не тільки Wi-Fi, але і Bluetooth та інших безпроводових технологій.

Одним із основних напрямів розвитку майбутніх мереж може стати створення передавача, що налаштовується, і який зможе підтримувати різні стандарти безпроводових комунікацій в одному діапазоні частот. Створення такого програмованого радіомодуля надасть змогу операторам мобільного зв’язку більш гнучко розміщувати базові станції.

Враховуючи зазначене можна стверджувати, що час вертикально-інтегрованих моделей, коли мережу будує, обслуговує і наповнює послугами один оператор, підходить до кінця. Майбутнє мереж – інфокомунікаційна кооперація.

Процес конвергенції різнорідних типів зв’язку має остаточно стерти грані між раніше жорстко розділеними послугами фіксованих і мобільних мереж. На шляху до цього необхідно вирішити не тільки технологічні, а і інші нормативні проблеми, але на цьому шляху стоять не стільки технологічні проблеми, скільки приналежність абонента, білінг, власність і т.д. Створення інфокомунікаційних кооперацій неможливе без вирішення цих проблем з метою забезпечення абонентів якісними послугами.

Для забезпечення цих вимог необхідно забезпечити роботу швидкодіючого Інтернету із широкою смугою доступу із високошвидкісним наскрізним підключенням, з оптимізованим протоколом, адресацією і маршрутизацією,

підтримкою відкритих загальних служб і додатків та з еволюційним підходом до архітектури мережі, яка має 100% покриття території.

1.2 Історія і актуальність напряму IoT

Концепція і термін для IoT вперше сформульовані засновником дослідницької групи Auto-ID (англ.) при Массачусетському технологічному інституті Кевіном Ештоном (англ. Kevin Ashton) в 1999 році на презентації для керівництва компанії Procter & Gamble. У презентації розповідалося про те, як всеосяжне впровадження засобів радіочастотної ідентифікації (RFID) зможе видозмінити систему управління логістичними ланцюгами в корпорації та дозволить порахувати і відстежити товари без людського втручання. У 2004 році в Scientific American була опублікована велика стаття, присвячена «інтернету речей», що наочно показує можливості концепції в побутовому застосуванні: в статті наведена ілюстрація, що показує як побутові прилади (будильник, кондиціонер), домашні системи (система садового поливу, охоронна система, система освітлення), датчики (теплові, датчики освітленості і руху) і «речі» (наприклад, лікарські препарати, що забезпечені ідентифікаційної міткою) взаємодіють один з одним за допомогою комунікаційних мереж (інфрачервоних, бездротових, силових і слабкострумових мереж) і забезпечують повністю автоматичне виконання процесів (включають кавоварку, змінюють освітленість, нагадують про прийом ліків, підтримують температуру, забезпечують полив саду, дозволяють зберігати енергію і керувати її споживанням). Самі по собі представлені варіанти домашньої автоматизації були не новими, але упор в публікації робився на об'єднанні пристроїв і «речей» в єдину обчислювальну мережу, яка обслуговується інтернет-протоколами. Тому розгляд «інтернету речей» як особливого явища сприяло набуттю концепцією широкої популярності. У звіті Національної розвідувальної ради США (англ. National Intelligence Council) 2008 року «інтернет речей» фігурує як одна з шести потенційно руйнівних технологій, вказується, що повсюдне і непомітне для споживачів перетворення в

інтернет-вузли таких поширених речей, як товарна упаковка, меблі, паперові документи, може завдати шкоди національній інформаційній безпеці. Період з 2008 по 2009 рік аналітики корпорації Cisco вважають «справжнім народженням "Інтернету речей"», так як, за їхніми оцінками, саме в цьому проміжку кількість пристроїв, підключених до глобальної мережі, перевищила чисельність населення Землі, тим самим «інтернет людей» став «інтернетом речей».

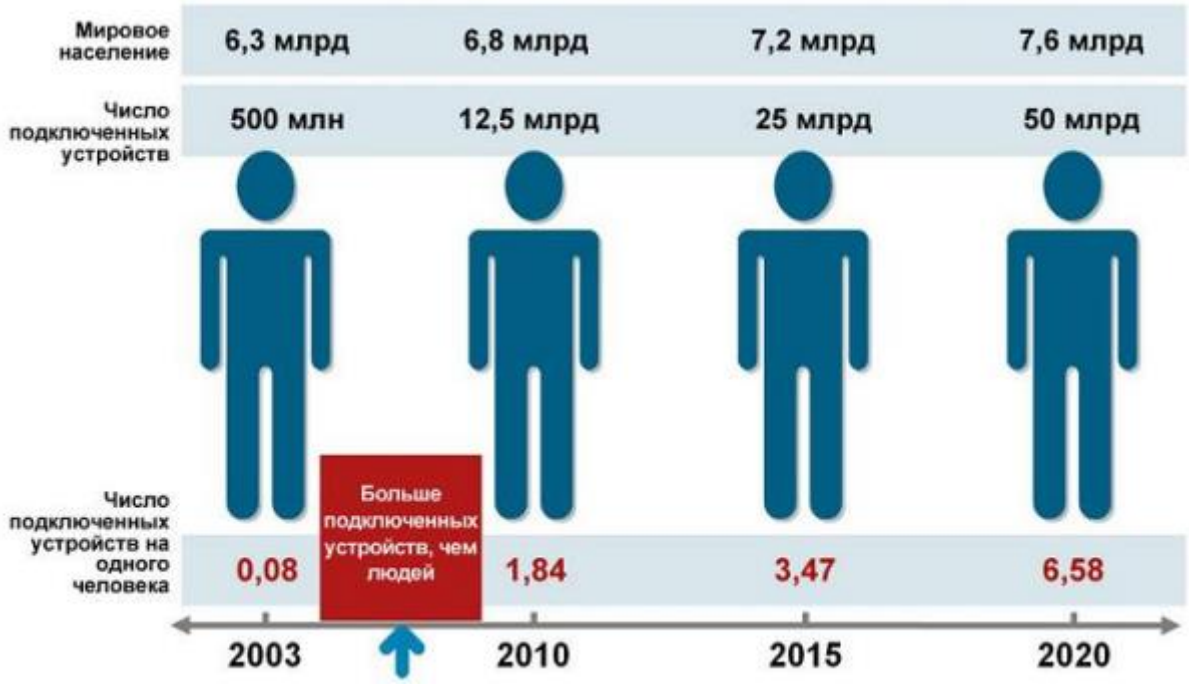

Рисунок 1.2 Кількість підключених пристроїв до глобальної мережі

З 2009 року за підтримки Єврокомісії в Брюсселі щорічно проводиться конференція «Internet of Things», на якій представляють доповіді єврокомісари і депутати Європарламенту, урядові чиновники з європейських країн, керівники таких компаній як SAP, SAS Institute, Telefónica, провідні вчені великих університетів і дослідницьких лабораторій.

З початку 2010-х років «інтернет речей» стає рушійною силою парадигми «туманних обчислень» (англ. Fog computing), що розповсюджує принципи хмарних обчислень від центрів обробки даних до величезної кількості взаємодіючих географічно розподілених пристроїв, яка розглядається як платформа «інтернету речей».

Починаючи з 2011 року Gartner поміщає «інтернет речей» в загальний цикл зрілості нових технологій на етап «технологічного тригера» із зазначенням терміну становлення більше 10 років, а в 2012 році випущений спеціальний цикл зрілості для технологій «інтернету речей».

За прогнозами Gartner, до 2020 року кількість підключених до всесвітньої мережі пристроїв становитиме 26 мільярдів, а дохід від продажу устаткування, програмного забезпечення та послуг становитиме 1,9 трлн. дол.. Деякі інші аналітичні агентства висловлюють ще більш оптимістичні прогнози. Найбільші світові IT компанії вже почали перегони за лідерство на цьому ринку. Так корпорація Intel у 2014 році після випуску «SoC Edison» оголосила конкурс «Make it Wearable» («Зробіть його одягненим») з призовим фондом $1,3 млн на найкраще застосування своєї системи для концепції IoT та створила власний підрозділ «Internet of Things Solutions Group» для розвитку цього напрямку. Компанія «Google» на початку 2014 року за 3,2 млрд доларів купила невелику фірму «Nest Labs», яка займається випуском інтелектуальних термостатів. Спеціалісти цієї компанії займались впровадженням на американському ринку технологій IoT. Виробники побутової техніки також працюють у цьому напрямку. Так на виставці CES 2014 у Лас-Вегасі була представлена велика кількість побутової техніки (холодильники, телевізори, пральні машини) з можливістю підключення до інтернет.

Значення на ринку прогнозується на рівні 80 мільярдів доларів. Лідерами у розробці та впровадженні інтернету речей є країни, в який розвинена індустрія виробництва мікропроцесорів та вбудованих комп'ютерів — це США, Китай, Південна Корея. Також значний прогрес у цій галузі демонструють європейські країни та Японія.

1.3 Висновки з розділу 1

При створенні безпроводових широкосмугових систем пріоритетом має стати:

– розробка нових технологій для гнучкого використання спектру та мобільного широко-смугового доступу, а також розробка концепції еталонних реалізацій з урахуванням комерційних і нормативних обмежень;

– топологія мереж має враховувати гнучкість, забезпечувати використання змішаних аналого-цифрових пристроїв і нових методів обробки сигналів, враховуючи необхідність її автономії, енергетичної ефективності при меншій потужності базових станцій, менших розмірів соти, високої пропускної здатності магістралей, високої електромагнітної сумісності. Мережі мають підтримувати велику кількість пристроїв, на багато порядків вище, ніж існуюча мережа Інтернету, обробку великої кількості потоків інформації, майбутні мережі мають створюватися на основі протоколу IP;

– інтеграція технологій радіозв'язку з волоконно-оптичними мережами, для об'єднання мобільних і бездротових мереж в комплексні системи зв'язку з метою забезпечення високошвидкісного бездротового доступу в усіх галузях діяльності людини;

– мережі мають бути готовими до забезпечення обміну нетипової для них інформації в нових областях застосування;

– забезпечення надійного захисту інформації, суміжний напрям – інфобанкінг, глобальний інформаційний депозитарій, система пов'язаних банківських даних, без яких не можуть існувати багато персонально орієнтовані сервіси;

– створення нової системи управління інфокомунікаційними мережами.

Впровадження повсюдного інтернету речей - це віддалена перспектива. Впровадження інтернету речей відбуваються не в глобальних масштабах, а всередині компаній. Технологія розумних речей здатна підвищити продуктивність праці в першу чергу в виробничому сегменті, логістичному бізнесі, транспортних і енергетичних компаніях.

РОЗДІЛ 2. АНАЛІЗ ТЕХНОЛОГІЙ ПЕРЕДАВАННЯ ІНФОРМАЦІЇ ДЛЯ ЗАСТОСУНКІВ ІНТЕРНЕТУ РЕЧЕЙ

2.1 Огляд сучасних технологій передавання інформації

Бездротові технології – це технології передачі інформації між об’єктами на відстані, тобто сигнал розповсюджується у відкритому середовищі.

Сигнал може бути в інфрачервоному, оптичному, радіочастотному діапазоні. На даний момент основними видами бездротових технологій, які використовуються в багатьох галузях діяльності суспільства, це Wi-Fi, Bluetooth, ZigBee, WiMAX. Ці технології мають багато загальних характеристик функціонування, але деякі з них принципово різні, як з технічної точк зору так і з сфери кінцевого призначення. Далі буде більш детально розглянуто технології Bluetooth, ZigBee, Wi-Fi.

2.1.1 Bluetooth

Технологія Bluetooth - виробнича специфікація бездротових персональних мереж (Wireless personal area network, WPAN). Bluetooth забезпечує обмін інформацією між такими пристроями, як персональні комп'ютери (настільні, кишенькові, ноутбуки), мобільні телефони, принтери, цифрові фотоапарати, мишки, клавіатури, джойстики, навушники, гарнітури на надійній, безкоштовної, повсюдно доступній радіочастоті для ближнього зв'язку. Bluetooth дозволяє цим пристроям повідомлятися, коли вони знаходяться в радіусі до 10 м один від одного (дальність сильно залежить від перешкод і завад), навіть у різних приміщеннях.

Своєму народженню *Bluetooth* зобов'язана фірмі *Ericsson*, що в 1994 році почала розробку нової технології зв'язку. Спочатку основною метою була розробка радіоінтерфейсу з низьким рівнем енергоспоживання й невисокою вартістю, що дозволяв би встановлювати зв'язок між стільниковими телефонами й бездротовими гарнітурами. Однак згодом роботи з розробки радіоінтерфейсу плавно переросли в створення нової технології.

Технологія Bluetooth підтримує як з'єднання типу «точка - точка», так і «точка – декілька точок». Два або більше пристроїв, що використовують один канал утворюють пікомережу (*piconet*). Один із пристроїв в такій мережі працює як основний (*master*), а інші - як залежні (*slave*). В одній піко-мережі може бути до семи активних залежних пристроїв, при цьому інші залежні пристрої перебувають у стані «паркування», залишаючись синхронізованими з основним пристроєм. Взаємодіючі пікомережі утворять «розгалужену мережу» (*scatternet*).

У кожної пікомережі діє тільки один основний пристрій, однак залежні пристрої можуть входити до різних пікомереж. Крім того, основний пристрій однієї пікомережі може бути залежним в іншій (Рисунок 2.1).

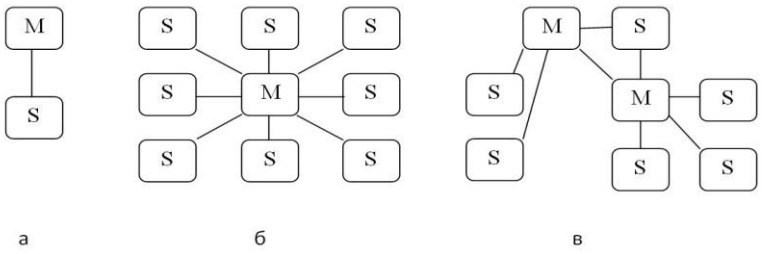

Рисунок 2.1 Види пікомереж

На рисунку 2.1 представлені 3 різновида пікомереж Bluetooth:

а) - з одним залежним пристроєм;

б) - з кількома залежними пристроями;

в) - розгалужена мережа

У більшості випадків технологія *Bluetooth* використовується розроблювачами для заміни провідного послідовного з'єднання між двома пристроями на бездротове. Для організації з'єднання й виконання передачі даних розроблювачеві необхідно програмно, за допомогою команд інтерфейсу хост-контролера реалізувати верхні рівні стека протоколу *Bluetooth*, до яких відносять*: L2CAP, RFCOMM, SDP*, а також профіль взаємодії по пос-лідовному порту - *SPP (Serial Port Profile)* і профіль виявлення послуг *SDP (Service Discovery Profile).*

Відомо, що *Bluetooth* і *Wi-Fi* використовують той самий неліцензійний діапазон 2,4 Ггц. Отже, у тих випадках, коли *Bluetooth* пристрої перебувають у зоні дії пристроїв *Wi-Fi* і здійснюють обмін даними між собою, можуть відбуватися колізії і це може вплинути на працездатність пристроїв. Технологія *AFH* дозволяє уникнути появи колізій: під час обміну інформа-цією для боротьби з інтерференцією технологія *Bluetooth* використовує стрибкоподібну зміну частоти каналу, при виборі якого не враховуються частотні канали, якими здійснюють обмін даними пристрої *Wi-Fi*.На рисунку 2.2 ілюструється принцип технології *AFH*.

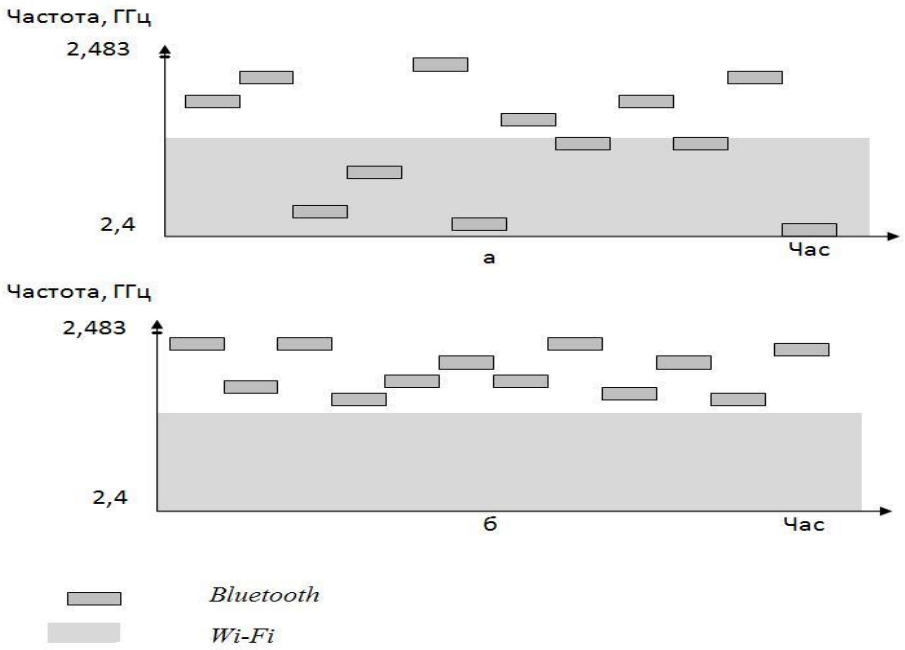

Рисунок 2.2 Принцип дії технології *AFH*. а – колізії; б - перехід від колізій за допомогою адаптивної перестройки частоти.

Для обміну даними за технологією *AFH* необхідно від 20 до 30 каналів (для обробки сигналів *Bluetooth* потрібно 79 каналів, кожний канал вимагає смуги частот 1 МГц). Таким чином, скоротивши необхідну кількість каналів можна зменшити використовуваний діапазон частот і уникнути пе-рекривання сигналів від пристроїв *Bluetooth* і *Wi-Fi[1]*.

2.1.2 ZigBee

Технологія бездротової передачі даних *ZigBee* була представлена на ри-нку вже після появи технологій бездротової передачі даних *Bluetooth* і *Wi-Fi*. Поява технології *ZigBee* обумовлено, насамперед, тим, що для деяких операцій (наприклад, для віддаленого керування освітленням або гаражними воротами, або зчитування інформації з датчиків) основними критеріями при виборі технології бездротової передачі є низьке енергоспоживання апаратної частини і її низька вартість. Цим обумовлена низька пропускна спроможність, тому що в більшості випадків живлення датчиків здійснюється від умонтованої батареї, час роботи від якої повинен перевищувати кілька місяців і навіть років. Існуючі на той момент часу технології бездротової передачі даних *Bluetooth* і *Wi-Fi* не відповідали цим критеріям, забезпечуючи передачу даних на високих швидкостях, з високим рівнем енергоспоживання й вартості апаратної частини. В 2001 році робочою групою № 4 *IEEE* 802.15 були розпочаті роботи зі створення нового стандарту, який би відповідав наступним вимогам: низький рівень енергоспоживання апаратної частини, що реалізує технологію бездротової передачі даних (час роботи від батареї повинен становити від декількох мі-сяців до декількох років); передача інформації повинна здійснюватися на не високій швидкості; низька вартість апаратної частини. Результатом ста-ла розробка стандарту *IEEE* 802.15.4[10]. У багатьох публікаціях під стандартом *IEEE* 802.15.4 розуміють технологію *ZigBee* і навпаки під *ZigBee* — стандарт *IEEE* 802.15.4. Однак це не так. Стандарт *IEEE* 802.15.4 визначає взаємодію тільки двох нижчих рівнів моделі взаємодії: фізичного рівня (*PHY*) і рівня керування доступом до радіоканалу для трьох неліцензійних діапазонів частот: 2,4 Ггц, 868 Мгц і 915 Мгц. У табл. 1.1 наведені основні характеристики устаткування, що функціонує в цих діапазонах частот.

Таблиця 1.1 Основні характеристики устаткування

Діапазон частот	Кількість каналів	Швидкість передачі даних	Тип модуляції
868 - 870 МГц	1	20Кбіт/с	*BPSK*
902 - 928 МГц	10	40Кбіт/с	*BPSK*
2.4 - 2.4835 ГГц	16	250Кбіт/с	*0-Q PSK*

У свою чергу, технологія бездротової передачі даних ZigBee, запропонована альянсом ZigBee, визначає інші рівні моделі взаємодії, до яких відносять мережний рівень, рівень безпеки, рівень структури додатка й рівень профілю додатка. Мережний рівень, технології бездротової передачі даних ZigBee, відповідає за виявлення пристроїв і конфігурацію мережі й підтримує три варіанти топології мережі, наведені на рисунку 1.4. В таких мережах може бути використаний режим синхронізованого доступу й «сну» мережійних пристроїв, що дозволяє зменшити енергоспоживання. Цей режим підтримують мережійні протоколи ZigBee. У таких мережах (Рисунок 1.4а) мережійні пристрої «слухають» ефір і «говорять» в ефір у моменти часу, «прив'язані» до сигналів маяків. В інший час пристрої «сплять». Може «спати» і пристрій, що випромінює сигнали маяків. Цю ситуацію можна трактувати як гомогенний (однорідний) розподіл потужності між всіма мережійними пристроями.

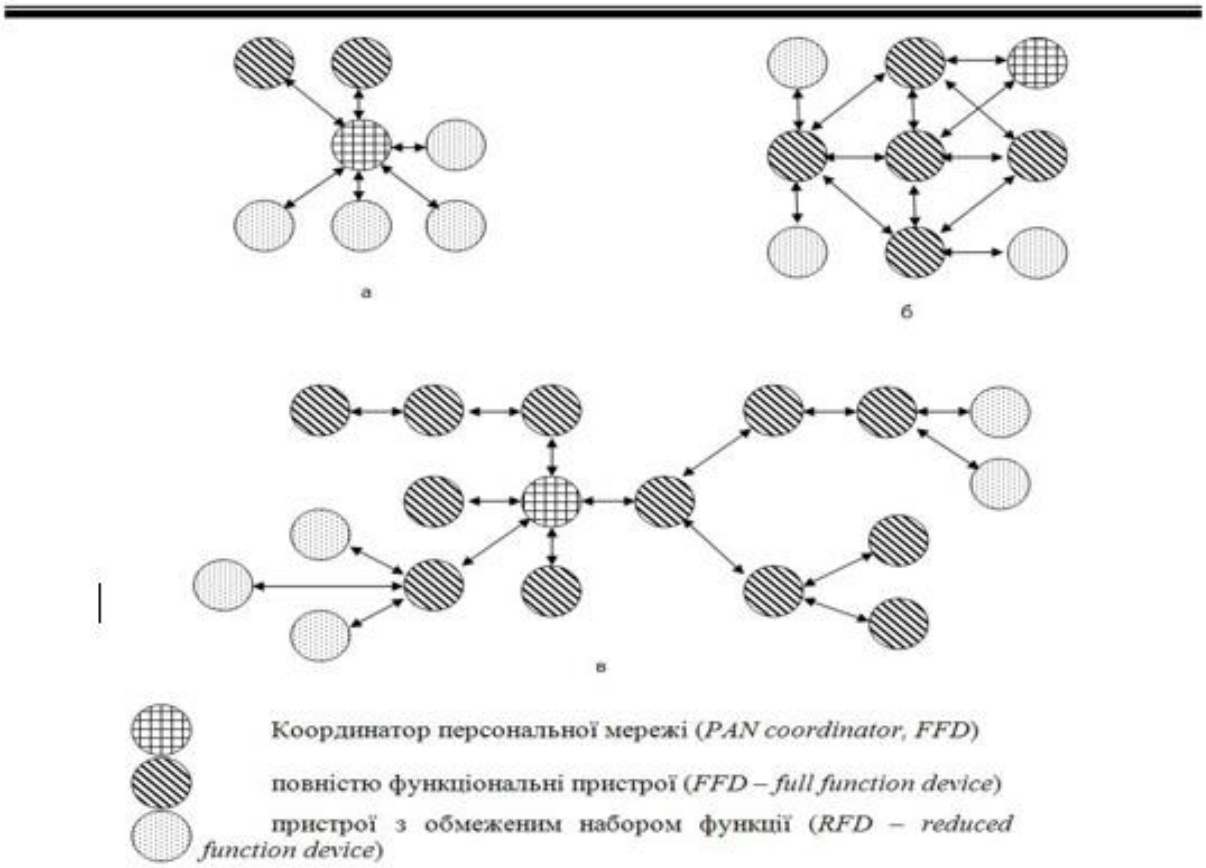

Рисунок 2.3 Варіанти топології мережі *ZigBee*

Мережі з топологією *Mesh* (Рисунок 2.3 б,в) принципово орієнтовані на асинхронну передачу даних, приміром, команд включення і вимикання в мережах керування або даних від «прокинувшихся» або «не сплячих» сен-сорів у сенсорних мережах. Вся мережа повинна «не спати» і бути завжди готова передати ці дані адресатові, приміром, приладам або центру збору інформації.

Для забезпечення низької вартості інтеграції технології бездротової передачі ZigBee у різні сфери фізична реалізація апаратної частини станда-рту IEEE 802.15.4 виконується у двох виконаннях: пристрої з обмеженим набором функцій (RFD) і повністю функціональні пристрої (FFD). При реалізації однієї з топологій мережі, наведеної на рисунку 1.4, потрібна наявність, принаймні, одного FFD пристрою, що виконує роль мережійного коорди-натора (МК). У табл. 2.2 наведений перелік функцій, виконуваних пристроями FFD і RFD.

Таблиця 2.2 - Перелік функцій для RFD та FFD пристроїв.

Пристрої з обмеженим набором функції (*RFD – reduced function device*)	Повністю функціональні пристрої (*FFD – full function device*)
При об'єднанні *RFD* пристроїв може використовуватися тільки топологія «зірка»	При об'єднанні *FFD* пристроїв можуть використовуватися усі можливі топології: «зірка», «кожен з кожним», «кластерне дерево».
Не можуть виступати в ролі МК	Можуть виступати в ролі МК, забезпечуючи при цьому маршрутеризацію повідомлень всередині мережі
Для обміну даними можуть встановлювати зв'язок тільки з МК (*FFD* пристроєм)	Обмін даними може проводитись з МК, іншим *FFD* пристроєм, чи *RFD* пристроєм.
Живляться, переважно, від вбудованої батареї	Живляться, переважно, від зовнішнього джерела

Технологія *ZigBee* може бути інтегрована у системи автоматизації життєзабезпечення будинків і будов (вилучене керування мережними розетка-ми, вимикачами, реостатами й т.д.); системи керування побутовою елект-ронікою; системи автоматичного зняття показань із різних лічильників (га-зу, води, електрики й т.д.); системи безпеки (датчики задимлення, датчики доступу й охорони, датчики витоку газу, води, датчики руху й т.д.); системи моніторингу навколишнього середовища (датчики температури, тиску, вологості, вібрації й т.д.); системи промислової автоматизації*[1]*.

2.1.3 Wi-Fi

Стандарти бездротової технології передачі даних для *Wi-Fi* є досить заплутаними, а тому спочатку варто визначити термінологію.

Стандарт *IEEE* 802.11 є базовим стандартом для побудови бездротових локальних мереж (*Wireless Local Network — WLAN*). Стандарт *IEEE* 802.11 постійно вдосконалювався, а тому зараз існує сімейство, до якого відносять специфікації IEEE 802.11 з буквеними індексами *a, b, c, d, e, g, h, i, j, k, I, m, n, o, p, q, r, s, u, v, w*. Однак тільки п'ять з них (*a, b, g, i* та *n*) є основними й користуються

найбільшою популярністю у виробників устаткування, інші ж являють собою доповнення, удосконалення або виправ-лення прийнятих специфікацій.

Для просування на ринку пристроїв для бездротових локальних мереж (*WLAN*) була створена група, що одержала назву Альянс *Wi-Fi*. Цей альянс здійснює керівництво роботами по сертифікації устаткування різних виробників і видачі дозволу на використання членами Альянсу *Wi-Fi* логотипа торговельної марки *Wi-Fi*. Наявність на устаткуванні логотипа *Wi-Fi* гарантує надійну роботу й сумісність устаткування при побудові бездротової локальної мережі (*WLAN*) навіть при використанні пристроїв різних виробників. На сьогоднішній день *Wi-Fi* сумісним є устаткування, побудоване по стандарту *IEEE* 802.11*a* [2], *b* [3] і *g* [4] (для забезпечення захищеного з'єднання також може використовуватися стандарт *IEEE* 802.11i [5]). Крім того, наявність на устаткуванні логотипа *Wi-Fi* означає, що робота устаткування здійснюється в діапазоні 2,4 Ггц або 5 Ггц. Отже, під *Wi-Fi* варто розуміти сумісність устаткування різних виробників, призначеного для побудови бездротових локальних мереж, з урахуванням викладених вище обмежень.

Перша специфікація стандарту *IEEE* 802.11, прийнята в 1997 році, установлювала передачу даних на швидкості 1 і 2 Мбіт/с у неліцензійному діапазоні частот 2,4 Ггц, а також спосіб керування доступом до фізичного середовища (радіоканалу), що використовує метод множинного доступу із упізнаванням несучої й усуненням колізій (*Carrier Sense Multiple Access with Collision Avoidance, CSMA-CA*).

IEEE 802.11*n* [6] - новітня версія стандарту 802.11 для мереж *Wi-Fi*. Цей стандарт був затверджений 11 вересня 2009. Стандарт 802.11*n* підвищує швидкість передачі даних практично вчетверо в порівнянні із пристроями стандартів 802.11*g* (максимальна швидкість яких дорівнює 54 Мбіт/с), за умови використання в режимі 802.11*n* з іншими пристроями 802.11*n*. Теоретично 802.11*n* здатний забезпечити швидкість передачі даних до 480 Мбіт/с. Пристрої 802.11*n* працюють у діапазонах 2,4 - 2,483 або 5,0 ГГц. Крім того, пристрої 802.11*n* можуть працювати в режимах:

1. наслідуваному (*Legacy*), у якому забезпечується підтримка пристроїв 802.11*b/g* і 802.11*a;*
2. змішаному (*Mixed*), у якому підтримуються пристрої 802.11*b/g*, 802.11*a* і 802.11*n;*
3. «чистому» режимі - 802.11*n* (саме в цьому режимі й можна ско-ристатися перевагами підвищеної швидкості й збільшеною дальністю передачі даних, забезпечуваними стандартом 802.11*n*).

Специфікація 802.11*n* передбачає використання як стандартних каналів шириною 20 МГц, так і широкосмугових - на 40 МГц із більш високою пропускною здатністю. Проект версії 2.0 рекомендує застосовувати 40-мегагерцові канали тільки в діапазоні 5 ГГц, однак користувачі багатьох пристроїв такого типу одержать можливість вручну переходити на широ-космугові канали навіть у діапазоні 2,4 ГГц. Ключовий компонент стандарту 802.11*n* за назвою MIMO (*Multiple Input, Multiple Output* - багато входів, багато виходів) передбачає застосування просторового мультиплексування з метою одночасної передачі декількох інформаційних потоків по одному каналі, а також багатопроменеве відбиття, що забезпечує доставку кожного біта інформації відповідному одержувачеві з невеликою ймовірні-стю впливу перешкод і втрат даних. Саме можливість одночасної передачі й прийому даних визначає високу пропускну здатність пристроїв 802.11*n*.

У табл. 2.3 наведені основні технічні характеристики стандартів *IEEE* 802.11*a*, *b* ,*g* і *n*.

Таблиця 2.3 - Основні характеристики стандартів

Стандарт	***IEEE 802.11a* [2]**	***IEEE 802.11b* [3]**	***IEEE 802.11g* [4]**	***IEEE 802.11n* [6]**
Частотний діапазон, ГГц	5.15-5.25 5.67-5.85	2.4-2.483	2.4-2.483	2.4-2.483 5.15-5.25 5.67-5.85
Доступ до радіоканалу	*CSMA-CA*	*CSMA-CA*	*CSMA-CA*	*CSMA-CA*
Кількість абонентів на один канал	64	64	64	64
Максимальна швидкість обміну даними	54Мбіт/с	11 Мбіт/с	54 Мбіт/с	480 Мбіт/с
Метод модуляції	*OFDM*	*BPSK,CCK*	*OFDM*	*BPSK, QPSK, 16-QAM, 64-QAM*
Дальність дії в приміщенні	10-20	20-100	20-50	10-20

Розроблювачі специфікації 802.11*n* подбали про те, щоб компоненти на її базі зберігали сумісність із пристроями стандарту 802.11*b* або 802.11*g* у діапазоні 2,4 Ггц та з пристроями 802.11*a* - у діапазоні 5 Ггц. У нових ме-режах 802.11*n* ще довгий час буде працювати безліч колишніх бездротових клієнтів, а тому при розгортанні бездротових ЛВМ (локальних високошвидкісних мереж) адміністраторові варто обов'язково передбачити їх підтримку*[1]*.

2.2 Класифікація бездротових технологій

Сучасні тенденції розвитку мереж доступу визначають жорсткі вимоги до смуги пропускання каналів “останньої милі”, до якості доставки сигналу і можливості надання мультисервісних послуг (Triple Play). Зростаючий попит абонентів на послуги, пов’язані з цифровим відео та швидкісним доступом до ресурсів мережі Internet накладають підвищені вимоги до смуги пропускання та гарантованого часу доставки пакетів.

Вже сьогодні в розвинутих країнах, орієнтованих, зокрема, на широкосмуговий безпроводовий доступ, в мережах, що охоплюють всі категорії

клієнтів, пропонується швидкість абонентських каналів 25-50 Мбіт/с від оператора до абонента і більше 10 Мбіт/с для висхідного трафіку (США, країни Азіатсько-Тихоокеанського регіона, Європа). Такі ж вимоги до смуги пропускання і для нових мереж у абонентів з’являються і в Україні, що демонструють пропозиції альтернативних операторів телекомунікацій. Розвиток технологій безпроводового доступу є відносно новим напрямом. Найбільш перспективними технологіями доступу з можливістю організації високошвидкісного каналу на сьогоднішній день є технології WiFi, 3G і WiMax. Впровадження WiFi може забезпечити потреби користувачів не лише по широкосмуговому доступу і Інтернет, але і в частині передачі мови поверх WiFi. Крім того, WiFi надає можливості по побудові цільових мереж, що може сприяти розширенню клієнтської бази і представленню користувачам принципово нових послуг.

Технології безпроводових технологій на ділянці доступу до мережі можна класифікувати, у першу чергу, по масштабах мережі зв'язку. На рисунку 1.5 наведено таку класифікацію.

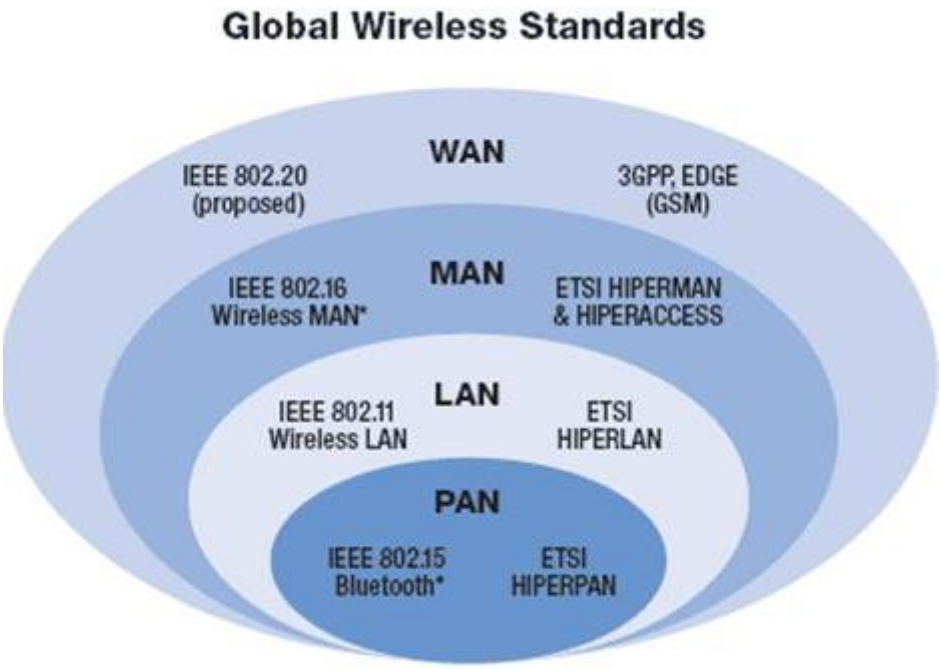

Рисунок 2.4 Зони покриття безпроводових мереж різного призначення

Починаючи з найменшої зони покриття можна виділити наступні групи мереж:

● PAN (Private Area Network) – персональна мережа. Прикладом може служити мережа, побудована в рамках одного приміщення, організована з використанням Bluetooth або WiFi;

● LAN (Local Area Network) – локальна мережа. Прикладом може служити мережа підприємства або організації;

● MAN (Metropolitan Area Network) – мережа в масштабах міста або населеного пункту;

● WAN (Wide Area Network) – глобальна мережа.

Основною метою створення безпроводових мереж доступу можуть бути: – забезпечення можливості переміщення абонента (мобільності); – зниження вартості мережі за рахунок виключення лінійного встаткування на абонентській ділянці мережі;

– створення мереж особливого призначення для рішення специфічних завдань, наприклад, у частині одержання телеметричної інформації (WSN);

– підвищення комфорту користувача шляхом виключення необхідності використання кабелів для з'єднання різних пристроїв. З погляду операторів зв'язку найбільший інтерес у цей час представляють мережі PAN, LAN і MAN. У цей час для реалізації мереж цих масштабів існують технології доступу WiFi і WiMax, а також технології доступу в мережах другого й третього поколінь. Безпроводові технології рівня абонентського доступу (АД) одержують у цей час все більше поширення. Інтенсивне поширення цих технологій почалося з розвитком мереж рухомого зв'язку.

2.3 Історія розвитку стандартів IEEE 802.11

Комітет зі стандартів IEEE 802 сформував робочу групу по стандартам для бездротових локальних мереж 802.11 в 1990 році. Було поставлено завдання по розробці загального стандарту для радіоустаткування і мереж, що працюють на

частоті 2,4 ГГц, зі швидкостями доступу 1 і 2 Mbps. Роботи зі створення стандарту були завершені через 7 років, і в червні 1997 року була ратифікована перша специфікація стандарту 802.11 для радіоустаткування і мереж, що працюють в частотному діапазоні 2402-2480 МГц, зі швидкостями доступу 1 і 2 Мбіт / с. Стандарт IEEE 802.11 був першим стандартом для продуктів WLAN від незалежної міжнародної організації, що розробляє більшість стандартів для дротових мереж. Однак на той час закладена спочатку швидкість передачі даних в бездротовій мережі вже НЕ задовольняла потреби користувачів. Для того, щоб зробити технологію Wireless LAN популярної, дешевої, а головне, що задовольняють сучасним жорстким вимогам бізнес-додатків, розробники були змушені створити новий стандарт.

У вересні 1999 року IEEE 802 ратифікувала розширення попереднього стандарту, назване IEEE 802.11b (також відоме, як 802.11 High rate), яке працює на швидкості 11 Мбіт/с (подібно Ethernet), що дозволило успішно застосовувати ці пристрої в великих організаціях. Сумісність продуктів різних виробників гарантується незалежною організацією, яка називається Wireless Ethernet Compatibility Alliance (WECA). Ця організація була створена лідерами індустрії бездротового зв'язку в 1999 році. З продуктами, які відповідають вимогам Wi-Fi (термін WECA для IEEE 802.11b), можна Ознайомитися на сайті WECA.

Пристрої Wi-Fi були призначені саме для корпоративних користувачів, щоб замінити традиційні кабельні мережі. Основний виграш такої заміни в тому, що собівартість прокладки мережі сильно скорочується за рахунок зменшення обсягів ручної роботи. Для провідної мережі потрібна ретельна розробка топології мережі і прокладка вручну багатьох сотень метрів кабелю, деколи в найнесподіваніших місцях. Для організації ж бездротової мережі потрібно тільки встановити в одній або декількох точках офісу базові станції (центральний приймач-передавач з антеною, підключений до зовнішньої мережі або сервера) і вставити в кожен комп'ютер мережеву плату з антеною. Основна робота фахівця-установника полягає в тому, щоб не залишалося "мертвих" зон в будівлі або на поверсі (залізобетонні перекриття екранують сигнал, і тоді на кожен поверх потрібна своя

станція). Після цього людей і комп'ютери можна переміщати як завгодно, навіть переїзд в новий офіс не зруйнує одного разу створену мережу.

Однак спочатку стандарт 802.11 замислювався як альтернатива саме Ethernet (тобто внутрішньоофісної зв'язку). Зрозуміло, він годився і для того, щоб виходити в Інтернет, якщо у комп'ютера, де встановлено такий пристрій, є вихід на зовнішню виділену лінію. Для передачі даних стандарт 802.11b використовує частоту 2,4 ГГц.

З усіх прототипів бездротової передачі, що мали хоч якусь перспективу, 802.11b, він же Wi-Fi, до пори до часу був найменш популярним. Великі компанії та економічно розвинені країни експериментували з WAP і GPRS і готувалися до продажу ліцензій на стільниковий зв'язок третього покоління (в стандарті UMTS), яка була покликана забезпечити високошвидкісну передачу даних в стільникових мережах. Широко обговорювалися можливості стандарту Bluetooth, за допомогою якого, як передбачалося, в мережу будуть об'єднані персональні пристрої і побутові прилади. Словом, альтернатив 802.11 було багато. Однак переміг все ж Wi-Fi.

2.4 Порівняння стандартів IEEE 802.11 на фізичному та канальному рівнях

2.4.1 Фізичний рівень

Для стандарту IEEE 802.11 на фізичному рівні визначені два широкосмугових радіочастотних метода передачі і один - в інфрачервоному діапазоні. Радіочастотні методи працюють в ISM діапазоні 2,4 ГГц і зазвичай використовують смугу 83 МГц від 2,400 ГГц до 2,483 ГГц. Технології широкополосного сигналу, що використовуються в радіочастотних методах, збільшують надійність, пропускну здатність, дозволяють багатьом непов'язаним один з одним пристроїв розділяти одну смугу частот з мінімальними перешкодами один для одного.

Стандарт 802.11 використовує метод прямої послідовності (Direct Sequence Spread Spectrum, DSSS) і метод частотних стрибків (Frequency Hopping Spread Spectrum, FHSS). Ці методи кардинально відрізняються, і несумісні один з одним.

Для модуляції сигналу FHSS використовує технологію Frequency Shift Keying (FSK). При роботі на швидкості 1 Mbps використовується FSK модуляція по Гаусу другого рівня, а при роботі на швидкості 2 Mbps - четвертого рівня.

Метод DSSS використовує технологію модуляції Phase Shift Keying (PSK). При цьому на швидкості 1 Mbps використовується диференціальна двійкова PSK, а на швидкості 2 Mbps - диференціальна квадратична PSK модуляція.

Заголовки фізичного рівня завжди передаються на швидкості 1 Mbps, в той час як дані можуть передаватися зі швидкостями 1 і 2 Mbps.

Метод передачі в інфрачервоному діапазоні (IR)

Реалізація цього методу в стандарті 802.11 заснована на випромінюванні ІК передавачем ненаправленного (diffuse IR) сигналу. Замість спрямованої передачі, що вимагає відповідної орієнтації випромінювача і приймача, який передається ІК сигнал випромінюється в стелю. Потім відбувається відображення сигналу і його прийом. Такий метод має очевидні переваги в порівнянні з використанням спрямованих випромінювачів, проте є і суттєві недоліки - потрібно стелю, що відображає ІК випромінювання в заданому діапазоні довжин хвиль (850 - 950 нм); радіус дії всієї системи обмежений 10 метрами. Крім того, ІК промені чутливі до погодних умов, тому метод рекомендується застосовувати тільки всередині приміщень.

На швидкості передачі даних 1 Mbps потік даних розбивається на квартети, кожен з яких потім під час модуляції кодується в один з 16-ти імпульсів. На швидкості 2 Mbps метод модуляції трохи відрізняється - потік даних ділиться на бітові пари, кожна з яких модулюється в один з чотирьох імпульсів. Пікова потужність сигналу, що передається становить 2 Вт.

Зміни, внесені 802.11b. Основне доповнення, внесене 802.11b в основний стандарт - це підтримка двох нових швидкостей передачі даних - 5,5 і 11 Mbps. Для досягнення цих швидкостей був обраний метод DSSS, так як метод частотних

стрибків у силу обмежень FCC не може підтримувати більш високі швидкості. З цього випливає, що системи 802.11b будуть сумісні з DSSS системами 802.11, але не будуть працювати з системами FHSS 802.11.

Для підтримки дуже зашумленних середовищ, а також роботи на великих відстанях, мережі 802.11b використовують динамічний зсув швидкості, який дозволяє автоматично змінювати швидкість передачі даних в залежності від властивостей радіоканалу. Наприклад, користувач може підключитися з максимальною швидкістю 11 Mbps, але в тому випадку, якщо підвищиться рівень перешкод, або користувач віддалиться на велику відстань, мобільний пристрій почне передавати на меншій швидкості - 5,5, 2 або 1 Mbps. У тому випадку, якщо можлива стійка робота на більш високій швидкості, мобільний пристрій автоматично почне передавати з більш високою швидкістю. Зрушення швидкості - механізм фізичного рівня, і є прозорим для вищих рівнів і користувача.

2.4.2 Канальний рівень

Канальний рівень 802.11 складається з двох підрівнів: управління логічним зв'язком (Logical Link Control, LLC) і управління доступом до носія (Media Access Control, MAC). 802.11 використовує той же LLC і 48-бітову адресацію, що і інші мережі 802, що дозволяє легко об'єднувати бездротові і дротяні мережі, однак MAC рівень має кардинальні відмінності.

MAC рівень 802.11 дуже схожий на реалізований в 802.3, де він підтримує безліч користувачів на загальному носії, коли користувач перевіряє носій перед доступом до нього. Для Ethernet мереж 802.3 використовується протокол Carrier Sence Multiple Access with Collision Detection (CSMA / CD), який визначає, як станції Ethernet отримують доступ до провідної лінії, і як вони виявляють і обробляють колізії, що виникають в тому випадку, якщо кілька пристроїв намагаються одночасно встановити зв'язок по мережі. Щоб виявити колізію, станція повинна мати здатність і приймати, і передавати одночасно. Стандарт

802.11 передбачає використання напівдуплексних приймачів, тому в бездротових мережах 802.11 станція не може знайти колізію під час передачі.

Щоб врахувати цю відмінність, 802.11 використовує модифікований протокол, відомий як Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), або Distributed Coordination Function (DCF). CSMA / CA намагається уникнути колізій шляхом використання явного підтвердження пакета (ACK), що означає, що приймаюча станція посилає ACK пакет для підтвердження того, що пакет отриманий неушкодженим.

CSMA/CA працює наступним чином. Станція, яка бажає передавати, тестує канал, і якщо не виявлено активності, станція чекає протягом деякого випадкового проміжку часу, а потім передає, якщо середовище передачі даних все ще вільна. Якщо пакет приходить цілим, приймаюча станція посилає пакет ACK, по прийомі якого відправником завершується процес передачі. Якщо передавальна станція не отримала пакет ACK, в силу того, що не була отримана пакет даних або прийшов пошкоджений ACK, робиться припущення, що сталася колізія і пакет даних передається знову через випадковий проміжок часу.

Для визначення того, чи є канал вільним, використовується алгоритм оцінки чистоти каналу (Channel Clearance Algorithm, CCA). Його суть полягає в вимірі енергії сигналу на антені і визначення потужності прийнятого сигналу (RSSI). Якщо потужність прийнятого сигналу нижче певного порогу, то канал оголошується вільним, і MAC рівень отримує статус CTS. Якщо потужність вище порогового значення, передача даних затримується відповідно до правил протоколу. Стандарт надає ще одну можливість визначення незайнятості каналу, яка може використовуватися або окремо, або разом з виміром RSSI - метод перевірки несучої. Цей метод є вибірковим, так як з його допомогою виробляється перевірка на той же тип несучої, що і за специфікацією 802.11. Найкращий метод для використання залежить від того, який рівень перешкод в робочій області.

Таким чином, CSMA / CA надає спосіб поділу доступу по радіоканалу. Механізм явного підтвердження ефективно вирішує проблеми перешкод. Однак він додає деякі додаткові накладні витрати, яких немає в 802.3, тому мережі

802.11 будуть завжди працювати повільніше, ніж еквівалентні їм Ethernet локальні мережі.

Інша специфічна проблема MAC-рівня - це проблема "прихованої точки", коли дві станції можуть обидві "чути" точку доступу, але не можуть "чути" один одного, в силу великої відстані або перешкод (рисунок 2.1).

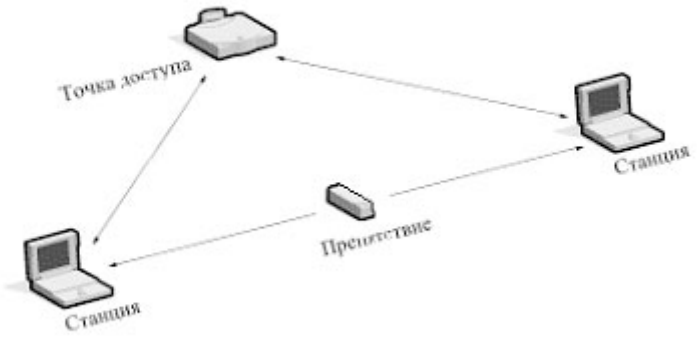

Рисунок 2.5 Проблема прихованої точки

Для вирішення цієї проблеми в 802.11 на MAC рівні доданий необов'язковий протокол Request to Send / Clear to Send (RTS / CTS). Коли використовується цей протокол, що посилає станція передає RTS і чекає відповіді точки доступу з CTS. Так як всі станції в мережі можуть "чути" точку доступу, сигнал CTS змушує їх відкласти свої передачі, що дозволяє передавальної станції передати дані і отримати ACK пакет без можливості колізій.

Нарешті, MAC рівень 802.11 надає можливість розрахунку CRC і фрагментації пакетів. Кожен пакет має свою контрольну суму CRC, яка розраховується і прикріплюється до пакету. Тут спостерігається відміну від мереж Ethernet, в яких обробкою помилок займаються протоколи більш високого рівня (наприклад, TCP). Фрагментація пакетів дозволяє розбивати великі пакети на більш маленькі при передачі по радіоканалу, що корисно в дуже "заселених" середовищах або в тих випадках, коли існують значні перешкоди, так як у менших пакетів менші шанси бути пошкодженими. Цей метод в більшості випадків

зменшує необхідність повторної передачі і, таким чином, збільшує продуктивність всієї бездротової мережі. MAC рівень відповідальний за збірку отриманих фрагментів, роблячи цей процес "прозорим" для протоколів більш високого рівня.

2.5 Висновки з розділу 2

В даному розділі висвітлюється порівняльна характеристика стандартів 802.11, зокрема на фізичному і канальному рівнях. Для стандарту IEEE 802.11 на фізичному рівні визначені два широкосмугових радіочастотних метода передачі і один - в інфрачервоному діапазоні. Для радіочастотного діапазону: метод прямої послідовності (Direct Sequence Spread Spectrum, DSSS) і метод частотних стрибків (Frequency Hopping Spread Spectrum, FHSS). Ці методи кардинально відрізняються, і несумісні один з одним. Для інфрачервоного діпазону є свій специфічний метод.

Проаналізовано канальний рівень для стандарту 802.11, і його підрівні:

1) управління логічним зв'язком
2) управління доступом до носія

Також освітленно основні пробеми MAC-рівня, зокрема проблему «прихованої точки». Для вирішення цієї проблеми в 802.11 на MAC рівні використовується необов'язковий протокол RTS / CTS.

РОЗДІЛ 3. ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ ОРГАНІЗАЦІЇ АРХІТЕКТУР ТА РІШЕНЬ ІНТЕРНЕТУ РЕЧЕЙ

3.1 Загальна топологія IoT рішення

На рисунку 3.1 представлена рівнева архітектура IoT рішень. Топологія IoT[8] відрізняється від звичайної рівневої моделі, такої як OSI. Це не лінійний і більш складний граф потоків. Деякі компоненти є необов'язковими і можуть бути відсутніми в конкретному класі рішень. Можуть присутні два типи логіки - M2M (від машини до машини) і M2P (від машини до людини), а також більш приватні випадки такі як C2C (від автомобіля до автомобіля, як правило в одній соті мобільного зв'язку LTE).

IoT рішення має два фізичних розташування - перший це кінцеві (периферійні) пристрої, а друге - в центрі обробки даних Backend на серверах або в хмарі. У той же час, це не класична архітектура клієнт-серверного додатка.

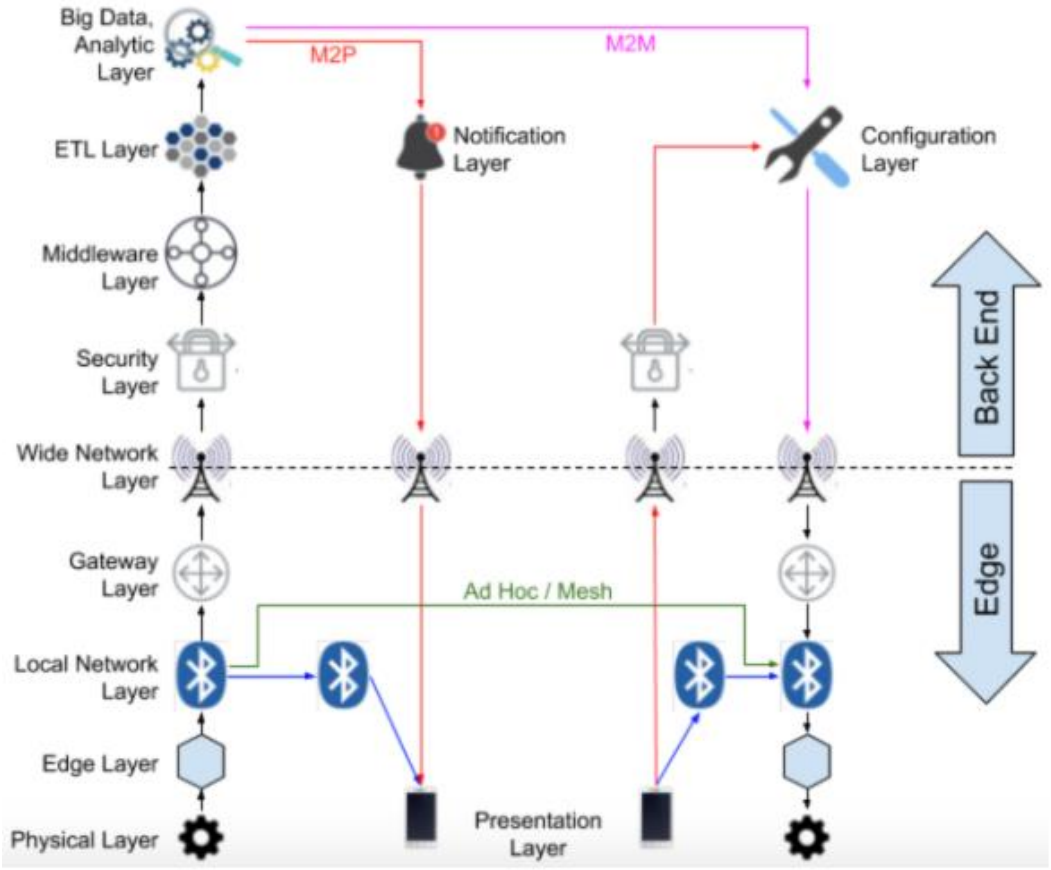

Рисунок 3.1 Рівнева архітектура IoT рішень

Physical Layer - фізичний рівень

Цей рівень являє два типи операцій - збір інформації (Датчики) і здійснення механічної роботи (виконавчі механізми).

Датчики можна розділити на наступні категорії:

- Сенсори:
    - Світлові: Фото діоди / транзистори / резистори, PIR детектори
    - Звукові: Мікрофони, ультразвукові сенсори
    - Вимикачі, зокрема кінцеві вимикачі, що реєструють крайні точки механічного руху. Вимірювачі кута повороту або швидкості обертання.
    - Електромагнітні сенсори вимірюють зміна фізичних характеристик, таких як електрична ємність, індуктивність, опір.
    - Складні або складові сенсори. До них відносяться спеціалізовані датчики, наприклад газу, спектра та ін., А також окремий вид пристроїв збору інформації отримує все більше застосування - Відео камери.

У рішеннях IoT фізичні елементи мають певні загальні вимоги:

- Як можна більш низьку ціну через великої кількості в рішенні IoT.
- Живлення від батарейки, що в свою чергу вимагає низького енергоспоживання. Сьогоднішній запит ринку - робота периферійних пристроїв без обслуговування від 1 до 10 років.
- Часто розташування в важкодоступних і віддалених місцях з мінімальними витратами на установку і обслуговування.
- У разі використання відеокамер, первинна обробка зображення з прийняттям рішення на основі штучного інтелекту

Виконавчі механізми IoT рішень відкривають замки вхідних дверей, пускають у хід двигуни, включають/вимикають світло, опалення, воду, газ та ін. Сильних змін до реалізації виконавчих механізмів не відбулося.

Можна підсумувати дві проблеми для вимог фізичного рівня:

- Низький рівень споживання енергії. Потрібен високий рівень інтеграції з верхніми шарами.
- Застосування відеокамер. Це також вимагає високого ступеня інтеграції з верхніми рівнями і вбудованими функціями AI/ML, реалізованими в периферійному пристрої.

Edge Layer - рівень периферійного обчислення

Цей рівень зазвичай підключається до одного датчику або виконавчого механізму. Він забезпечує мінімальну функціональність для перетворення аналогової інформації в цифрову і/або навпаки. Для підключення датчиків існують ті ж вимоги за ціною і споживаної потужності. Багато виробників, що випускають ці типи пристроїв, не мають єдиного стандарту для моделі даних, конфігурації і експлуатації, що створює окремі проблеми інтеграції.

Для зниження енергоспоживання периферійні пристрої зазвичай мають чотири режими роботи:

1. Режим сну
2. Режим вимірювання та збору інформації з датчиків
3. Режим зв'язку, передачі і отримання інформаці
4. Режим установки і підключення

Нижче на рисунку 3.2 представлена блок-схема периферійного пристрою.

Периферійний пристрій зазвичай об'єднує три рівні: фізичний, периферійного обчислення і комунікаційний. Основна функціональність рівня периферійного вичісленія- локальна ETL (Extract, Transformation and Load) - отримання, перетворення і збереження інформації з датчиків. Цей рівень є відповідальним не тільки за збір інформацію з датчика, але також і за приведення її до стандартного вигляду, фільтрацію перешкод, попередній аналіз і локальне збереження.

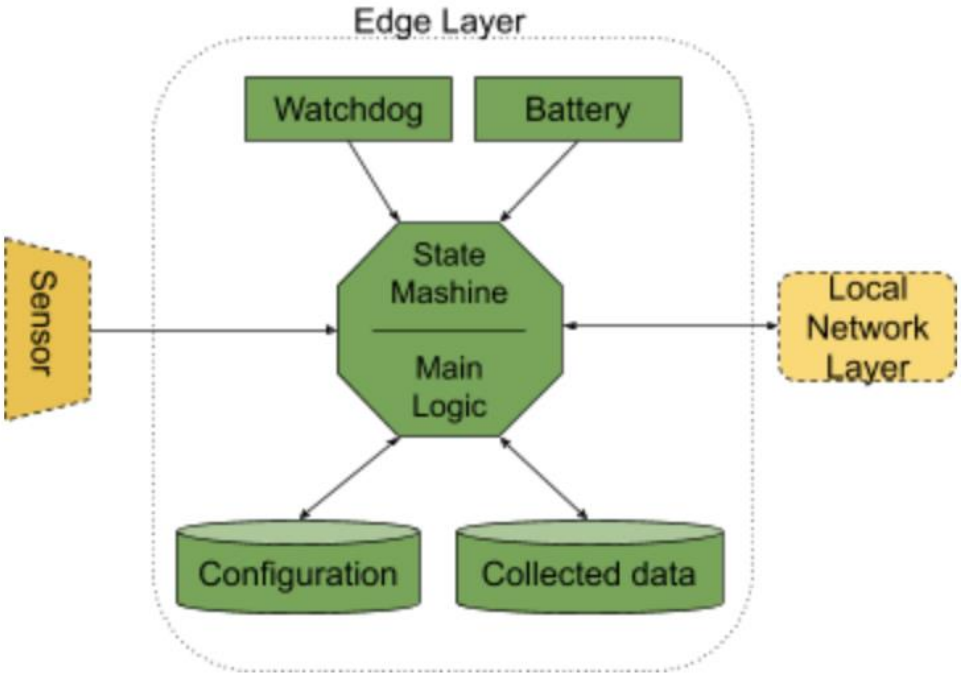

Рисунок 3.2 Блок-схема периферійного пристрою

Отже, основні вимоги рівня периферійного обчислення:

Низький рівень споживання енергії. Це може бути досягнуто за допомогою апаратного забезпечення з низьким енергоспоживанням і алгоритмів Sleep/WakeUp. Часто наявність локального елемента штучного інтелекту.

Local Network Layer - рівень периферійної комунікації

Передача даних є найбільш енергоємною частиною периферійного пристрою, тому що більшість периферійних пристроїв не підключені до електромережі й провідним засобам зв'язку. Крім того, периферійні пристрої можуть бути розташовані досить далеко від Шлюзу (в межах декількох кілометрів). З іншого боку, кількість переданої інформації зазвичай досить мало. Наступні протоколи використовуються на рівні периферійної комунікації:

- ZigBee / Zwave
- BLE
- LoRa
- Proprietary low band

Для збільшення відстані і надійності зв'язку Ad Hoc і Mesh широко використовуються сьогодні на цьому рівні.

Для цілей конфігурації також може використовуватися протокол NFC. У процесі першої установки і/або технічного обслуговування сервісний інженер з мобільним додатком може підключатися до периферійних пристроїв через рівень периферійної комунікації. Іноді Q-код, надрукований на периферійному пристрої, також використовується для аутентифікації.

Gateway Layer - рівень шлюзу

У IoT-рішенні існує кілька причин наявності рівня шлюзу:

- Якщо Backend буде отримувати необроблену інформацію, це збільшить його потужність і витрати будуть дуже великі.
- Робота Backend не може гарантувати реакцію в реальному часі для великої кількості периферійних пристроїв.
- Через обмеження безпеки деяка інформація не може бути відправлена в Backend і не може постійно контролюватися людиною. До такої інформації відносяться дані камер вуличного спостереження, медична інформація, тощо.

Шлюз повинен забезпечувати такий основний функціонал:

- Здійснювати другий рівень ETL від своїх периферійних пристроїв.
- Фіксувати критичну ситуацію і видавати локальну реакцію, навіть без зв'язку з BackEnd.
- Комунікувати BackEnd. Відправляє на сервер оброблену інформацію з периферійних пристроїв і отримує дані конфігурації для периферійних пристроїв.
- Зберігати інформацію про статус периферійних пристроїв, і дані ними зібрані.

У деяких випадках функціональність AI/ML (штучний інтелект/машинне навчання) має бути присутня на рівні шлюзу. Шлюзовий пристрій в основному харчується від електромережі або має велику вбудовану батарею, але в деяких рішеннях також потрібно низьке енергоспоживання. У такій ситуації виникає додаткова проблема - протокол синхронізації для зв'язку з периферійним пристроєм. Один з них (шлюз або периферійний пристрій) повинен передавати повідомлення «Готовий до спілкування» частіше, ніж інший пристрій готове вийти на зв'язок. Вибір буде залежати від загального енергоспоживання кожного пристрою і необхідного часу без обслуговування.

Wide Network Layer - рівень зовнішньої зв'язку

Цей шар розділяє периферійну і BackEnd частини загального рішення. Шлюз в основному підключений до BackEnd з використанням мобільного бездротового зв'язку, такий як 4G/5G, але іноді використовується бездротовий доступ до Інтернету. Логічний рівень зовнішньої зв'язку має стандартизований протокол для рішень IoT, який називається LvM2M. Протокол LvM2M був розроблений для доступу до кожного периферійного пристрою, але оскільки багато постачальників периферійних пристроїв не підтримують інтерфейси LvM2M, шлюзовий пристрій може вирішити цю проблему і створити обгортку для зв'язку з периферійними пристроями.

Рівень зовнішньої зв'язку містить також комунікаційні сервіси і моделі ISO всередині себе. Він включає служби балансування щоб визначити своє місцезнаходження, засновані на DNS сервісі, транспортний протокол COAP, шифрування DTLS і багато інших компонентів.

Security Layer - рівень безпеки

Цей рівень забезпечує функції AAA (Authentication, Authorization and Accounting - аутентифікація, авторизація та облік) і шифрування/дешифрування разом з іншими послугами, пов'язаними з Інтернетом. Всі Cloud-рішення мають

свої власні реалізації безпеки, але функціонально вони все побудовано на принципі ролей і дозволів.

Підключення кінцевого користувача до Backend також має компонент рівня безпеки.

Middleware Layer - рівень всередині серверного зв'язку

Цей рівень забезпечує внутрішню Cloud функціональність балансування навантаження, черги повідомлень і передачі потокової інформації. Компоненти цього шару повинні бути дубльовані і автоматично масштабуватися. Рівень реалізується в основному на основі мікросервісов або PaaS від Cloud провайдерів. Така вимога випливає з парадигми стрибків і провалів обсягу даних. Автоматичне масштабування знижує вартість Backend реалізації. Фактична реалізація сервісу може бути різною, але загальний принцип залишається одним - забезпечити асинхронну передачу повідомлень з буферизацією і перерозподілом навантаження. Таким чином різні компоненти Backend можуть виконувати свою роботу незалежно і горизонтально масштабироваться в залежності від навантаження.

Etl layer - рівень збору, обробки та зберігання даних

Внутрішній рівень ETL (витяг, перетворення і завантаження) є третьою операцією ETL. Перший був у периферійному пристрої, другий - в шлюзі. Back End ETL накопичує дані з усіх периферійних пристроїв і шлюзів і відповідає за такі операції:

- Збір інформації
- Приведення інформації до стандартного вигляду
- Збереження інформації для подальшого використання
- Управління життєвим циклом інформації включаючи архівування та знищення
- Повідомлення інших сервісів про надходження нових даних

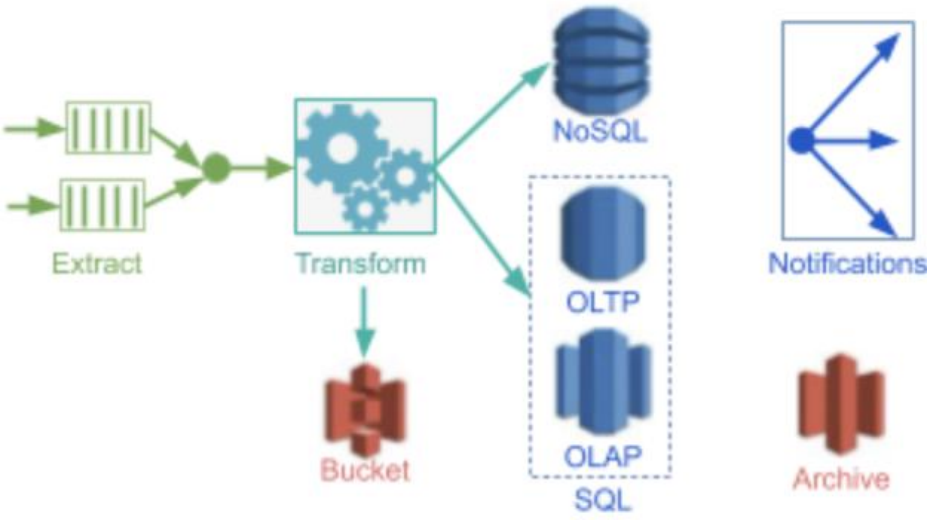

Рисунок 3.3 Загальна схема реалізації Etl-рівня

Загальна схема реалізації цього рівня представлена на рисунку 3.3. Операція збору даних (Extract) включає в собі читання інформації з релевантних черг. Операція трансформації може виконуватися спеціалізованими сервісами Cloud, такими як Ламбда, або обчислювальними засобами всередині контейнерів і просто віртуальними машинами. Кожен з перерахованих вище методів має свої позитивні і негативні властивості. Так наприклад, Ламбда сервіс зручний майже повною автоматизацією, але має чимало часу створення і тому не застосуємо, якщо потрібна швидка реакція на що з'явилися події. Також Ламбда погано підходить для постійних обробок, оскільки тарифікується за часом використання. Найбільш часто застосовна служба - контейнеризованих обчислення. Вони зручно масштабуються і легко переносяться на різні BackEnd. Основне завдання цієї операції - зробити приведення даних до зручного для зберігання, сортування і пошуку виду. Для цього часто дані об'єднуються з різних повідомлень і навіть черг.

Операції зберігання (Load) призначені для збереження, сортування і подальшого пошуку інформації. Залежно від типу інформації та варіантів її використання, застосовуються різні інструменти. Якщо дані не мають суворої схеми (колонок таблиці), то вони зберігаються в NoSQL базах. Однак, якщо дані можуть бути систематизовані фіксованою схемою, то використовуються SQL типи баз даних. Останні, в свою чергу мають 2 типу - OLTP (Online Transactional

Processing) і OLAP (Online Analytic Processing). Як випливає з назви, перший тип більш підходить для самого процесу ETL - записи в базу нових значень, в той час як другий зручніше для пошуку та аналізу даних. Тому часто після завантаження в OLTP базу, у фоновому режимі, дані копіюються в OLAP. Трапляються ситуації, коли дані не зручно або неможливо зберігати в базах даних, наприклад вигляді запису, Ці дані записують в Bucket, а метадані записів зберігають в базах даних. Для скорочення витрат на зберігання, застарілі дані архівуються або видаляються. І останнім компонентом цього рівня є внутрішня нотифікація про наявність нових збережених даних для подання клієнтам і для сервісів аналізу.

Big Data and Analytic Layer - рівень аналітики

Залежить від конкретного додатка IoT. Великі дані і аналітичний рівень отримуватимуть ситуативну інформацію з усього набору периферійних пристроїв. Ця частина менш стандартизована, тому що вона сильно відрізняється від однієї програми до іншого в силу різних завдань рішень. Алгоритми AI/ML також широко використовуються в цьому рівні.

Окремою категорією є передбачення майбутніх подій, таких як необхідні частини на складі, споживання майбутніх ресурсів, погода та ін.

Notification layer - рівень повідомлення

На цьому рівні може існувати кілька компонент, але всі вони мають алгоритм повідомлення за підпискою. Клієнтську програму підписується на необхідні події і, коли це відбувається, отримує інформаційний сигнал - повідомлення. В основному це програми електронної пошти і мобільні клієнти, меньше телефонні дзвінки (використовується для екстреного оповіщення). Мобільний додаток змушений переходити в сплячий режим для енергоспоживання, але ОС iOS і Android мають механізм повідомлень, що вказує на прибуття нових даних.

Presentation Layer - рівень представлення

Додаток IoT може мати два потоки: M2M (від машини до машини) і M2P (від машини до людини). Рівень представлення, пов'язаний з потоком M2M, де Back End обробляє інформацію і надає її клієнту або інженеру служби підтримки. Сьогодні не існує стандартизованого UI/UX уявлення для цього рівня.

Рівень представлення також відповідає за обслуговування, конфігурацію і зміни стану системи включаючи периферійні пристрої та шлюзи. У ньому представлені і команди на керуючі виконавчими механізмами периферійних пристроїв.

Configuration Layer - рівень конфігурацій

Цей рівень відноситься до обох потоків - M2M і M2P і працює як сховище для трьох типів статусів периферійних пристроїв:

- Актуальний стан периферійного пристрою
- Новий стан периферійного пристрою, який буде завантажено.
- Проміжний статус периферійного пристрою - вказує на процес оновлення від старих станів до нових. Часто цей статус відсутній.

Периферійний пристрій і навіть шлюз можуть мати тільки короткий час підключення до Backend. Будь-яка зміна статусу від клієнта або системи, зберігаються в цьому рівні, і протягом часу зв'язку відправляється на шлюз або периферійний пристрій.

Щоб така логіка працювала, зазвичай реалізується наступний процес комунікації (рисунок 3.4):

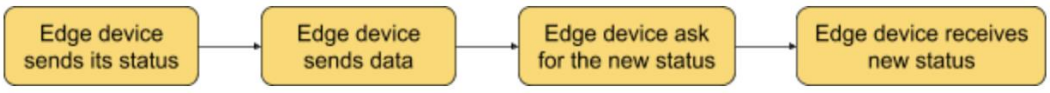

Рисунок 3.4 Процес комунікації

Якщо шлюз присутній в схемі передачі інформації, то більша частина інформації від периферійних пристроїв відправляється на серверну частину у вигляді пакетів даних, зібраних з кількох периферійних пристроїв.

3.2 Шаблони взаємодії компонент в мережах IoT

Перш ніж почати новий IoT-проект, варто подумати про те, які шаблони обміну інформацією найкращим чином для нього підійдуть. Насправді, прийняти це рішення слід якомога раніше, ще до того, як обрані протоколи, способи зв'язку та допоміжна інфраструктура, що розробляється. В основі цієї рекомендації лежить одна проста причина: не прийнявши таке рішення на самому початку, розробник, у міру розвитку проекту, ризикує сам себе загнати в кут, вибратися з якого можна буде лише серйозно переробивши код, архітектуру, модель безпеки рішення, і то, як воно взаємодіє із зовнішнім світом.

Шаблон «запит-відповідь»

Шаблон «запит - відповідь» (request - response), це, ймовірно, самий широко відомий шаблон обміну інформацією. Його реалізація передбачає наявність клієнта, який виконує запити до деякої служби, розташованої на сервері, що надає послуги. Сервер ще називають респондентом або відповідачем.

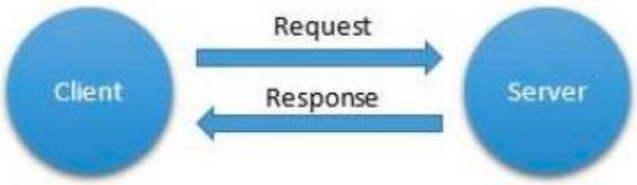

Рисунок 3.5 Схема взаємодії: шаблон «запит-відповідь»

Саме цей шаблон в основному використовує протокол HTTP. Він же є основою сервісно-орієнтованих архітектур, веб-служб і REST-рішень. Це практичний шаблон, особливо, якщо архітектура проекту передбачає наявність клієнтських і серверних частин або провідних і ведених сутностей.

Крім HTTP, шаблон «запит - відповідь» підтримують такі протоколи, як Constrained Application Protocol (CoAP) і Extensible Messaging and Presence Protocol (XMPP).

Основний недолік даного шаблону полягає в нерівності учасників обміну даними, що цілком очевидно проявляється в топології інтернету. Двохнаправлений обмін даними, коли обидва учасники запитують дані один у одного, може бути складний в реалізації, особливо якщо на шляху даних є мережеві екрани.

Плануючи використовувати шаблон «запит - відповідь» в проекті, потрібно вирішити, які частини системи будуть клієнтами, а які - серверами. Якщо, наприклад, якийсь датчик буде клієнтом, а IoT-шлюз - сервером, датчик сам буде вирішувати, коли йому передавати на сервер власні данні. Сервер, якщо йому знадобляться відомості з датчика, самостійно їх запросити не зможе. Якщо ж датчик зробити сервером, а шлюз - клієнтом, датчик можна буде опитувати коли завгодно. Однак, тут є одна проблема: якщо датчик недостатньо захищений, хто завгодно зможе до нього підключитися. Якщо ж в подібному рішенні задіяна надійна система безпеки, то, як наслідок, ускладниться спосіб взаємодії клієнта і сервера, та й сама система в цілому. Можливо, в проект потрібно буде додати додаткові служби, датчики доведеться оснащувати більш потужним апаратним забезпеченням. Крім того, всім цим буде складніше управляти.

Шаблон «підписка на події»

Цей шаблон (event subscription) дозволяє клієнтові підписуватися на події заданого типу на сервері. Сервер сповіщає клієнта щоразу, коли відбувається його цікавить подія. Як результат, відпадає необхідність в постійному опитуванні сервера.

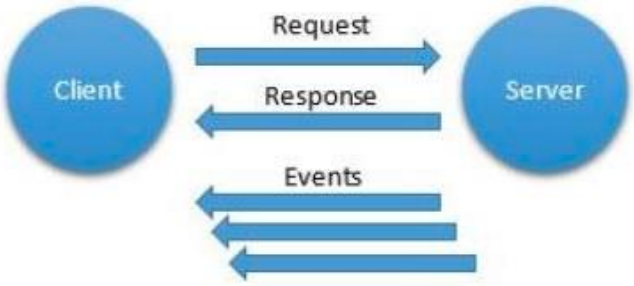

Рисунок 3.6 Схема взаємодії: шаблон «підписка на події»

Високий рівень механізм підписки на події може включати в себе вимоги, залежні від клієнта, що стосуються того, які саме події і за яких умов його цікавлять. Переваги використання підписки на події перед раніше розглянутим шаблоном «запит - відповідь», полягають в тому, що для обміну даними між клієнтом і сервером потрібно приблизно в два рази менше повідомлень. Крім того, дані передаються клієнту при виникненні якоїсь події, а не за запитом, що знижує до мінімуму час між виникненням якоїсь ситуації, що цікавить клієнта, і моментом, коли він про це дізнається.

Протоколи, які підтримують цей шаблон, включають в себе CoAP, XMPP і General Event Notification Architecture, який є частиною архітектури Universal Plug and Play, що базується на HTTP.

Шаблон «асинхронний обмін повідомленнями»

Асинхронний обмін повідомленнями (asynchronous messaging) передбачає можливість відправки повідомлень між рівноправними системами, що знаходяться на одному щаблі ієрархії. Цей шаблон має на увазі двонаправлений обмін повідомленнями.

Рисунок 3.7 Схема взаємодії: шаблон «асинхронний обмін повідомленнями»

Якщо використовуваний протокол підтримує асинхронний обмін повідомленнями, на його основі можна побудувати будь-які інші шаблони передачі даних.

Серед протоколів, які підтримують даний шаблон, можна назвати XMPP, Advanced Message Queuing Protocol (AMQP), і, на рівні IP - User Datagram Protocol

(UDP). Однак, у випадку з застосуванням UDP для реалізації цього шаблону, можливі проблеми з мережевими екранами.

Шаблон «надійна доставка повідомлень»

Додаткам, які виконують критично важливі функції, необхідно знати, що повідомлення було доставлено одержувачу як мінімум один раз. Власне кажучи, при використанні шаблону асинхронного обміну даними ця вимога виконується. Повідомлення може загубитися в дорозі, але використання шаблону «запит - відповідь» дозволяє подати запит, повідомлення знову, до тих пір, поки не буде отримано підтвердження (або відповідь) від сторони, яка повинна повідомлення отримати. Тут потрібно враховувати, що загубитися може і повідомлення, і відповідь про його отриманні, а тому цей шаблон гарантує, що повідомлення буде доставлено як мінімум один раз. Однак, те, що повідомлення буде доставлено одержувачу не більше одного разу (або не менше ніж один раз), не дуже підходить деяким додаткам, наприклад, таким, які задіюють концепцію транзакцій або виконують підрахунок повідомлень.

Застосування шаблону надійної доставки повідомлень (reliable messaging) дає гарантію того, що повідомлення буде доставлено одержувачу в точності один раз. Серед протоколів, що підтримують надійну доставку повідомлень, можна відзначити Message Queuing Telemetry Transport (MQTT), AMQP. Крім того, завдяки відкритим розширенням, подібний функціонал підтримують HTTP і XMPP

Шаблон «багатоадресна передача повідомлень»

Попередній шаблон зайнятий обміном повідомленнями між двома об'єктами. Іноді, однак, потрібно більш ефективний підхід, якщо одну і ту ж інформацію потрібно, в один і той же час, відправити декільком одержувачам. Найпростіший з шаблонів, що реалізують подібний функціонал - це «многоадресная передача повідомлень» (multicasting). В рамках цього шаблону відправник відсилає одне повідомлення через проміжну ланку системи (це може

бути брокер або маршрутизатор), після чого повідомлення пересилається кільком одержувачам, кожен з яких зареєструвався для отримання подібних повідомлень.

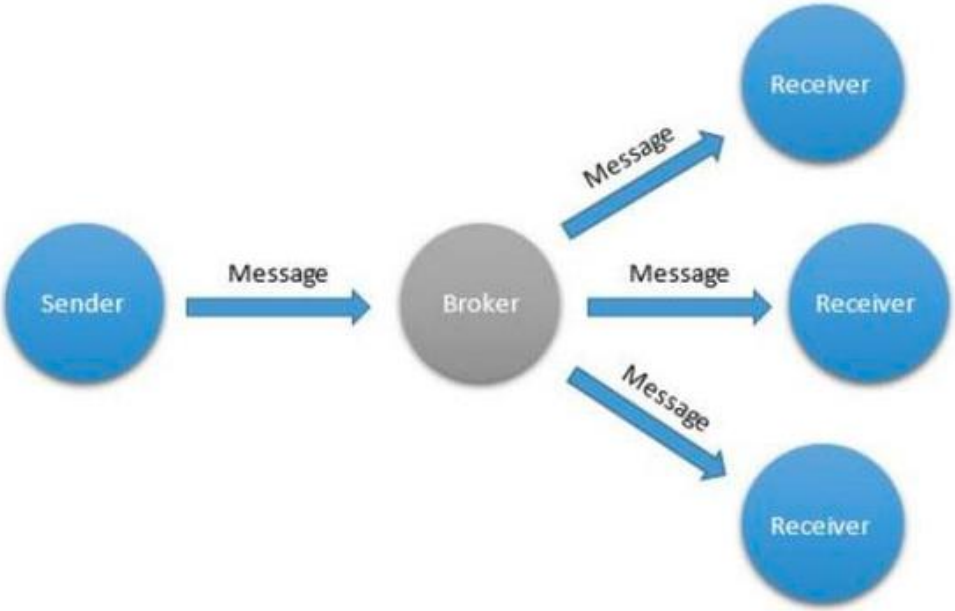

Рисунок 3.8 Схема взаємодії: шаблон «багатоадресна передача повідомлень»

Завдяки використанню даного шаблону можна знизити навантаження на мережу, так як відправнику не потрібно самостійно надсилати одне й те саме повідомлення кожному, хто його чекає. Насправді, відправнику навіть не потрібно знати, хто саме отримає повідомлення. Цей шаблон може бути вельми корисний у безлічі ситуацій. Наприклад, при синхронізації безлічі пристроїв або при розподілі одних і тих же даних між декількома одержувачами. Багатоадресну передачу повідомлень підтримують протоколи XMPP, AMQP і UDP.

Тут доречно висловити деякі застереження. Стосуються вони використання багатоадресної передачі повідомлень для реалізації інших схем зв'язку.

Так, хоча цей шаблон і можна використовувати для зниження навантаження на мережу, часто до нього звертаються як до способу обходу обмежень у використовуваному протоколі, а також - для реалізації на базі якогось протоколу шаблону «підписка на події». Якщо, наприклад, використовувати багатоадресну передачу повідомлень для того, щоб зменшити затримки в мережах, де потрібно, але неможливо, реалізувати шаблон «підписка на події», даний шаблон призведе

ні до зниження, а до підвищення навантаження на мережу. Крім того, систему з багатоадресною передачею даних складніше захистити.

Що стосується підвищення ефективності використання пропускної здатності мережі при використанні многоадресной передачі даних, то ресурси можна заощадити лише в тому випадку, якщо одержувачі споживають більшу частину отриманих даних. Якщо ж значна частка переданих таким чином даних не використовується одержувачами - це привід розглянути інші шаблони взаємодії.

Шаблон «видавник - підписник»

Шаблон «видавець - підписник» (publish - subscribe) є розширенням шаблону многоадресной доставки повідомлень. Принципова різниця між ними полягає в тому, що надіслані повідомлення зберігаються на проміжному вузлі. Ці повідомлення, або посилання на них, потім розподіляються по зацікавленим в них передплатникам.

Особливості реалізації шаблону залежать від використовуваного протоколу, залежить від нього, а також від налаштувань підвузлів, і те, які саме повідомлення зберігаються. Це може бути тільки найсвіжіше повідомлення, або задану кількість повідомлень, або все повідомлення.

Тут, крім того, важлива різниця в передачі самого повідомлення і посилання на нього, так як це впливає на необхідну смугу пропускання мережі, і, як результат, на продуктивність рішення.

Якщо підписники використовують більшість повідомлень, то передача самих повідомлень більш ефективна, як і в випадку з багатоадресною передачею даних. Якщо ж фактичне споживання даних одержувачами залежить від деяких додаткових факторів, то ефективніше передавати посилання на повідомлення. Вони менші, ніж самі повідомлення, а підписники, найімовірніше, використовують лише невелике їх число для того, щоб отримати ті повідомлення, на які вказують посилання. У подібному випадку, для того, щоб отримати повідомлення по посиланню, потрібно виконувати додаткові звернення до вузла зберігання повідомлень за моделлю «запит - відповідь».

Шаблон «видавець - підписник» підтримують такі протоколи, як MQTT, AMQP, XMPP.

Шаблон «черга»

Черги, а конкретно - черги FIFO - це шаблон обміну даними, який дозволяє одній або більшій кількості сутностей відправляти якісь повідомлення або завдання для обробки в чергу, після чого один або кілька одержувачів отримують ці повідомлення в тому порядку, в якому вони були поставлені в чергу .

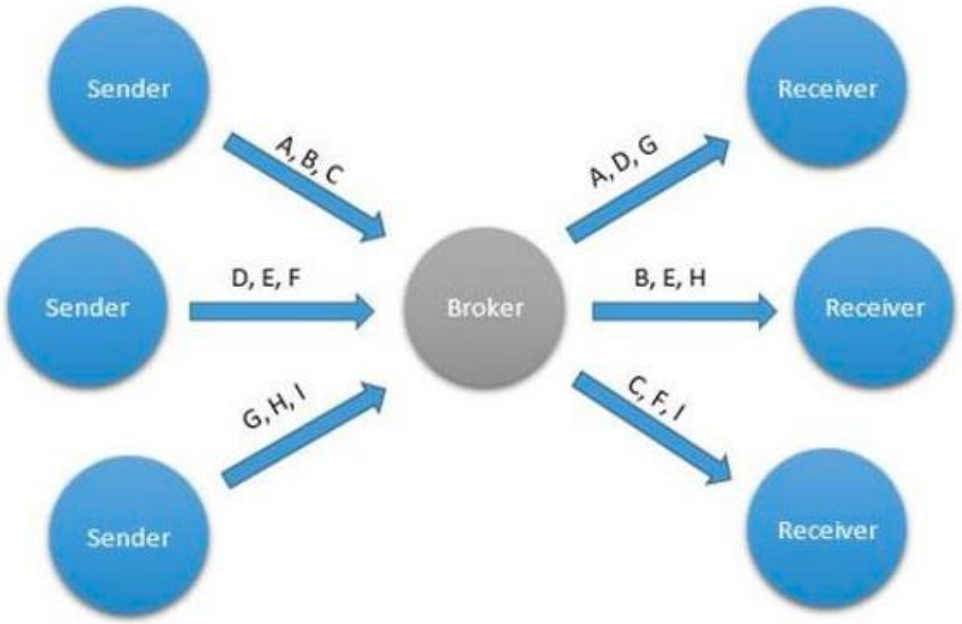

Рисунок 3.9 Схема взаємодії: шаблон «черга»

Черга зазвичай знаходиться на проміжному вузлі, або в мережі, до якої підключені всі учасники обміну даними. Черги - це чудовий засіб для балансування навантаження. У чергу збирають завдання з різних джерел і розподіляють їх серед існуючих обробників, можливо, володіють різною продуктивністю.

Використовуючи чергу, можна уникнути жорсткого зв'язку між системами, які передають дані, і системами, які ці дані отримують і обробляють. В результаті, в залежності від реального робочого навантаження на систему, можна збільшувати

або зменшувати кількість приймачів і передавачів даних. Серед протоколів, про які вже згадувались раніше, лише AMQP володіє вбудованою підтримкою черг.

Шаблон «брокери повідомлень»

Брокери повідомлень (message brokers) зазвичай є стандартизованими компонентами допоміжної мережевої інфраструктури IoT-проектів. Вони красиво вирішують проблеми, викликані обмеженнями, які накладають мережеві екрани на двонаправлений обмін даними між пристроями. Брокер дозволяє сутностей підключатися до нього, займаючись передачею повідомлень між підключеними до нього клієнтами. Так як всі підключення виконані через брокера, тільки брокер повинен бути доступний з інтернету. Мережевому екрану не потрібно приймати або перенаправляти вхідні підключення до пристроїв, як було б потрібно при використанні протоколу, що забезпечує зв'язок рівноправних систем, жорстко обмеженого подібною моделлю обміну повідомленнями.

Крім управління повідомленнями, брокери можуть надавати підключеним клієнтам додаткові служби. Наприклад, брокер може виступати посередником при реалізації шаблону багатоадресного обміну повідомленнями, шаблонів «видавець - підписник» і «чергу».

Крім того, брокери повідомлень зазвичай надають служби аутентифікації клієнтів. Це полегшує роботу в розподілених мережах, де перевірка справжності пристроїв може виявитися непростим завданням. Таким чином, якщо брокер може повідомляти про статус вже аутентіфіцированний учасників системи, включених в обмін даними, інші учасники можуть використовувати цю інформацію для прийняття рішень в сфері безпеки. Це, до того ж, позбавляє від необхідності реалізації власної схеми аутентифікації на кожного учасника обміну даними.

Хоча обмін повідомленнями між рівноправними системами - це лише один з варіантів організації зв'язку, подібні рішення повинні передбачати аутентифікацію клієнтів. Інакше серйозно постраждає безпеку системи. Якщо ж використовується протокол, який включає в себе брокери повідомлень, то, швидше за все, не

знадобиться розробляти власні допоміжні служби, які дозволять рішенням працювати надійно і безпечно.

Протоколи XMPP, AMQP і MQTT, в тій чи іншій формі, задіють цей шаблон.

Шаблон «федерація»

«Федерація» (federation) - це важливий шаблон, в якому якась глобальна мережа розбивається на логічні частини. Це дозволяє здійснювати глобальне масштабування рішення і забезпечує все необхідне для його природного зростання.

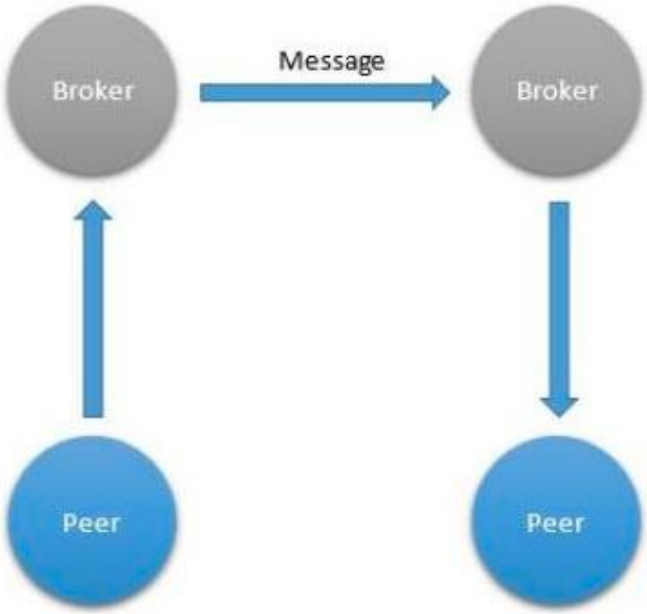

Рисунок 3.10 Схема взаємодії: шаблон «Федерація»

Основна ідея тут - дозволити збільшувати розміри рішення, не обмежуючи при цьому продуктивність наявної мережевої інфраструктури, використовуючи підхід «розділяй і володарюй».

При здійсненні зв'язку без брокерів, наприклад, як при використанні протоколів HTTP і CoAP, федеративна структура є на рівні домену. Кожен домен вказує на власний набір IP-адрес, з ним пов'язаний власний веб-сервер. У систему

можна додавати нові веб-сервера, в нових доменах, не обмежуючи доступ до існуючих систем. Такий підхід - один з основних ключів до успіху Всесвітньої павутини.

При використанні протоколів, які передбачають наявність брокерів і підтримують федерації, брокери з'єднуються між собою для маршрутизації повідомлень. Кожен брокер управляє аутентифікацією у власному домені і знає, як підключатися до інших доменів для перенаправлення в них повідомлень. Крім того, федеративні мережі з брокерами надають зручне рішення проблеми глобальної ідентифікації учасників обміну даними.

Найбільш широко відомий протокол, який використовує брокерів і федерації - Simple Mail Transfer Protocol (SMTP). Серед протоколів, про які ми говоримо в цьому матеріалі, які вміють працювати з брокерами, федерацію підтримує лише XMPP.

Шаблон «виявлення»

Розглянемо умовний приклад. Нехай, у нас є якесь виготовлене на заводі пристрій, який планується використовувати в IoT-системі. Якщо його, наприклад, планується застосовувати в системі з багатоадресною передачею даних, ми відразу ж зіткнемося з деякими складнощами, пов'язаними з інтеграцією пристроїв в систему.

Полягають вони в тому, що «речі» інформовані тільки про власну ідентифікаційну інформацію (це може бути щось на зразок MAC-адреси), але нічого не знають ні про те, як вони будуть «видно» в мережі, до якої їх планується підключити , ні про якийсь головний мережевий пристрій, з яким їм доведеться взаємодіяти.

Після установки і настройки (чим більше автоматизована настройка - тим краще), «речі» дізнаються про свою мережеву ідентифікаційну інформацію, але не про те, як підключитися до головного пристрою. У свою чергу, головного пристрою відомий власний мережевий адресу, а також - заводські дані «речей»

(які, наприклад, можна швидко ввести в систему, відсканувавши наклейки на коробках), але не мережеві дані інших пристроїв.

Шаблон обміну повідомленнями «виявлення» (discovery) дозволяє створити механізм, за допомогою якого проводиться зіставлення мережевих ідентифікаційних даних підлеглих пристроїв з мережевими даними головного вузла. Робиться це з використанням загальних знань про вихідні ідентифікаційних параметрах підлеглих пристроїв.

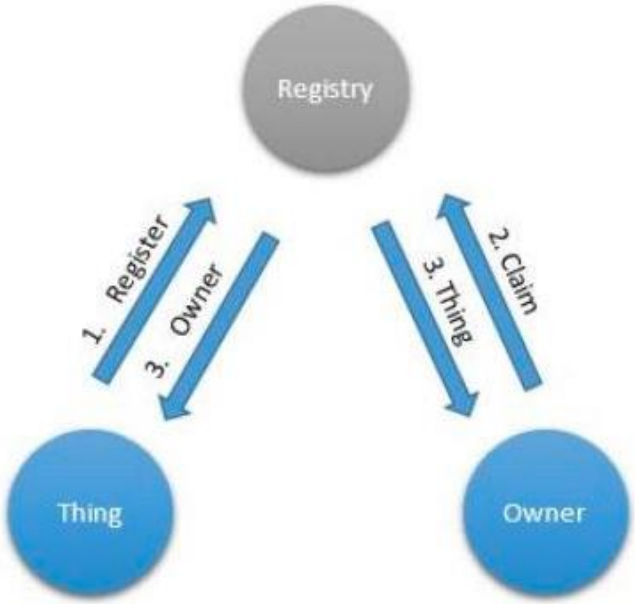

Рисунок 3.11 Схема взаємодії: шаблон «виявлення»

Даний шаблон реалізується з використанням «реєстру речей» (Thing Registry), доступного по мережі як самим «речам», так і головному пристрою. Клієнти реєструються в реєстрі, а головний пристрій, які мають до них через реєстр, використовуючи лише їх заводські ідентифікатори. Якщо запит успішний, то мережеві ідентифікаційні дані кожного з учасників обміну даними відправляються іншому, і обидва, таким чином, знають, як один з одним взаємодіяти. Існує розширення XMPP, яке підтримує цей шаблон.

Шаблон «делегування довіри»

В інтернеті важлива можливість прийняття виважених рішень в області безпеки. При використанні шаблону «делегування довіри» (delegation of trust) кінцеві пристрої перенаправляють запити до більш надійно захищеною, довіреної системи в реальному часі, а після отримання відповіді виконують якісь дії.

Рисунок 3.12 Схема взаємодії: шаблон «делегування довіри»

Дії довіреної сутності, при надходженні нових запитів від клієнтських систем, можуть бути засновані, наприклад, на машинному навчанні, або на налаштуваннях, які задає адміністратор, можливо, реагуючи на запити системи при надходженні їй нових запитів.

Для того, щоб можна було реалізувати цей шаблон, необхідно використовувати асинхронний двонаправлений обмін повідомленнями. Існує розширення XMPP, яке підтримує делегування довіри.

3.3 Практична реалізація шаблону «запит-відповідь»

Для практичної реалізації шаблону «запит-відповідь» створенно рішення та розгорнуто його на базі платі NodeMCU ESP8266 згідно схеми на рисунку 3.13.

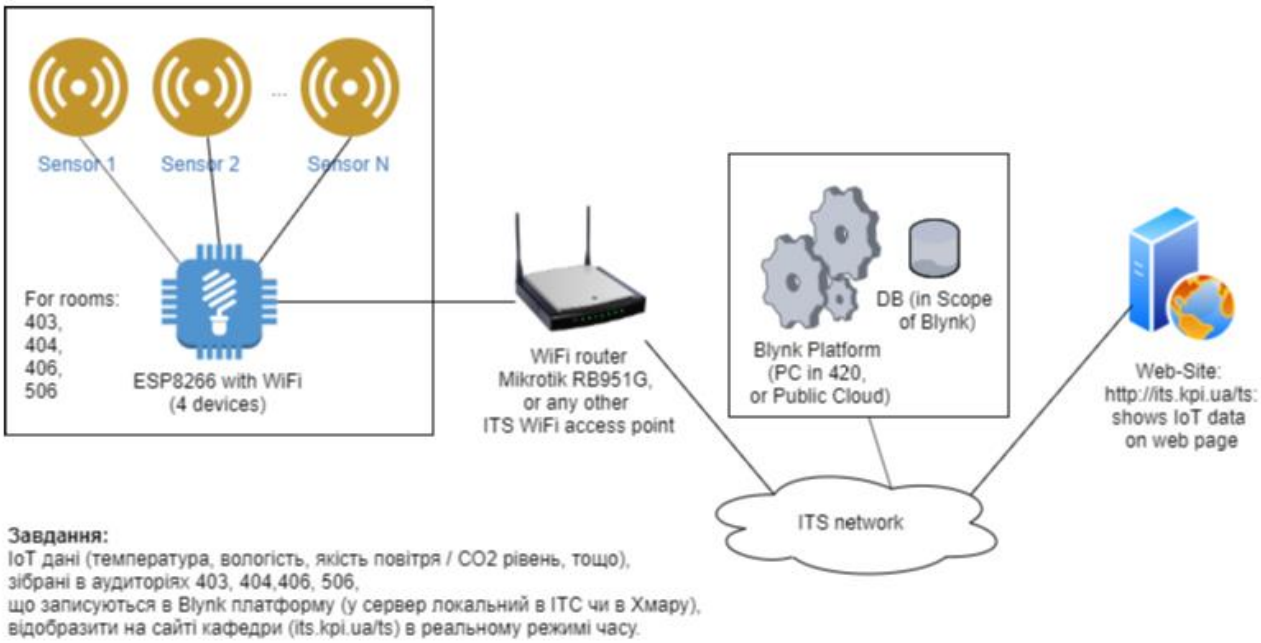

Рисунок 3.13 Схема мережі датчиків на кафедрі ТС

Для реалізації даної схеми потрібно виконати наступні етапи:

1. Модифікувати скетч для плати.
2. Написання і імпламентація серверу для отримання/обробки та зберігання даних з датчиків.
3. Створення iframe вікна на сторінці кафедри ТС сайту ІТС для відображення інформації з датчиків.

В якості основного пристрою обрана WiFi Плата **NodeMCU V3 ESP8266 (CP2102)**

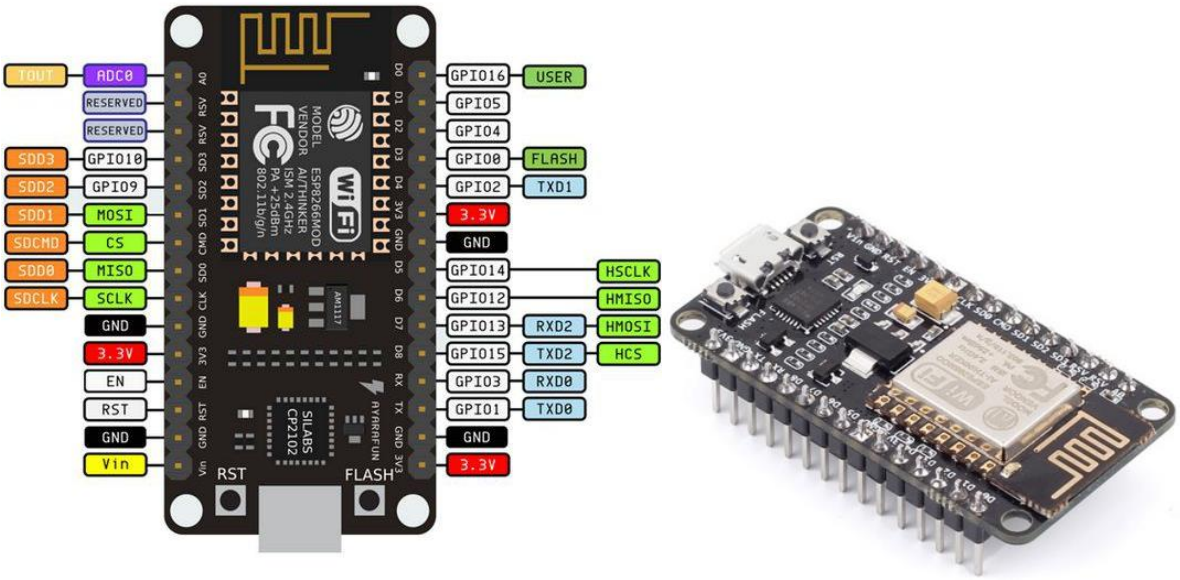

Рисунок 3.14 Плата NodeMCU V3 ESP8266

NodeMCU є платою розробника на базі чіпа ESP8266 (версія ESP12E), який представляє собою UART-WiFi модуль з ультра низьким споживанням. Сам чіп проектувався для пристроїв зі світу інтернет речей, а дана плата дозволяє спростити розробку, тому що на ній вже реалізовано підключення по USB, регулятор живлення і все виведення чіпа розведені на гребінки зі стандартним кроком 2.54 мм, що дозволяє вставити його в макетну плату і створити прототип навіть не включаючи паяльник. Крім цього плата поставляється з прошивкою NodeMCU, що дозволяє програмувати її за допомогою мови Lua або за допомогою Arduino IDE.

Характеристики:

• WiFi стандарту 802.11 b / g / n

• підтримка STA / AP / STA + AP режимів

• вбудований стек протоколів TCP / IP з підтримкою множинних клієнтських підключень (до 5)

• D0 - D8, SD1 - SD3: можуть бути використані як GPIO, PWM, IIC, тощо.

• струм на виведення: 15 мА

• AD0: 1 виведення АЦП

• живлення: 4.5 - 9 В (10 В максимум), 5В живлення від USB з наданням отладочного інтерфейсу

• споживання: обмін даними: 70 мА (200 мА максимум), очікування: <200 мкА

• швидкість передачі: 110 - 460800 б / сек

• підтримка UART / GPIO інтерфейсів передачі даних

• перепрошивка з хмари або через USB

• Розміри плати 48х26мм

• діапазон робочих температур: -40 - +125 град.С

• маса: 18 г

Макетна плата обрана MB-102 на 830 отворів. Велика макетна плата (830 отворів) з двома лініями для живлення з кожного боку (200 отворів). Дана макетная дозволяє без пайки створювати попередні електронні проекти для їх тестування, перевірки і налагодження перед остаточним монтажем. Розміри беспаечное макетної плати MB102 165х56 мм, кількість отворів 830.

Модуль живлення для макетної плати. Має увігнуту конструкцію і не затуляє собою робочу зону макету. Є перемикач 3.3 / 5В. Живиться від зовнішнього блоку живлення 6,5 - 12В. На платі є USB вихідний роз'єм для живлення 5В пристроїв з USB роз'ємом (наприклад контролерів Arduino).

Характеристики:

• Вхідна напруга: 6.5В - 12В (DC) або від USB

• Вихідна напруга: 3.3В / 5В (перемикається)

• Максимальний вихідний струм: <700 мА

• Відстань між шинами живлення: 42мм

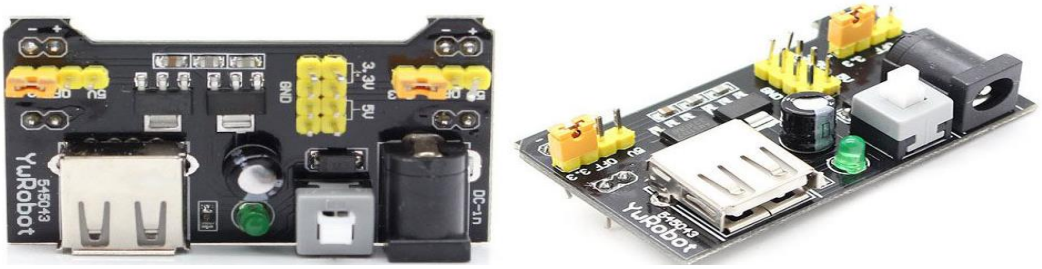

Рисунок 3.15 Живлення для плати

Датчик

Датчик DHT11 - це цифровий датчик температури і вологості, що дозволяє калібрувати цифровий сигнал на виході. Складається з ємнісного датчика вологості і термістора. Також, датчик містить в собі АЦП для перетворення аналогових значень вологості і температури.

живлення I / O 3.5-5.5 В

Характеристики:

• Визначення вологості 20-95% з 5% точністю

• Визначення температури 0-50 град. з точністю 2 град.

• Частота опитування не більше 1 Гц (не більше одного разу в 1 сек.)

• Розміри 15.5мм х 12мм х 5.5мм

• 4 виведення з відстанню між ніжками 0.1 "

Виходи:

1. Vcc (3-5в живлення)

2. Data out - Висновок даних

3. Не використовується

4. Загальний

Для можливості відправлення даних датчика з плати потрібно створити скетч для неї. В якому використовуються бібліотеки для роботи з датчиком DHT11

та бібліотека ESP8266WiFi.h для роботи з мережею WiFi. Вихідний код з коментаріями приведено нижче:

```
/*********
// підключаємо бібліотеку «ESP8266WiFi»:
#include <ESP8266WiFi.h>
#include "DHT.h"
#define DHTTYPE DHT11     // DHT 11

// вписуємо тут дані для своєї WiFi-мережі:
const char* ssid = "TS_ITS";
const char* password = "KafTSITS";

// веб-сервер на порті 80:
WiFiServer server(80);

// датчик DHT:
const int DHTPin = 5;
//ініціалізуєм датчик DHT:
DHT dht(DHTPin, DHTTYPE);

// тимчасові змінні:
static char celsiusTemp[7];
static char fahrenheitTemp[7];
static char humidityTemp[7];

// цей блок буде запускатися тільки при завантаженні ESP:
void setup() {
  // инициализируем послідовний порт (в налагоджувальних цілях):
Serial.begin(115200);
  delay(10);

  dht.begin();

  //підключаємося до WiFi-мережі:
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
```

```
  }
  Serial.println("");
  Serial.println("WiFi connected");
            //  "Підключення до WiFi виконано"

  // запускаєм веб-сервер:
  server.begin();
  Serial.println("Web server running. Waiting for the ESP IP...");
             // "Веб-сервер запущено. Ждем IP-адрес ESP..."
  delay(1000);

  // IP-адрес ESP:
  Serial.println(WiFi.localIP());
}

// цей блок буде запускатися знову і знову:
void loop() {
    WiFiClient clientH;
   const char* host = "esp8266temp.radio-signal-telecom.in.ua";
   if (!clientH.connect(host, 80)) {
    Serial.println("connection failed");
   } else {
      float h = dht.readHumidity();
          // зчитуємо температуру в Цельсіях (за замовчуванням):
          float t = dht.readTemperature();
          // зчитуємо температуру в Фаренгейтах
          // (isFahrenheit = true):
          float f = dht.readTemperature(true);
     Serial.print("GET /site_test_esp/?");

    Serial.print("push_data=true&c=");
    Serial.print(t);
    Serial.print("&h=");
    Serial.print(h);
    Serial.print("&f=");
    Serial.print(f);
    clientH.print( "GET /site_test_esp/?");

    clientH.print("push_data=true&c=");
    clientH.print(t);
    clientH.print("&h=");
    clientH.print(h);
    clientH.print("&f=");
    clientH.print(f);
```

```
    clientH.println( " HTTP/1.1");
    clientH.print( "Host:" );
    clientH.println(host);
    clientH.println( "Connection: close" );
        clientH.println();
        clientH.println();
   delay(10000);
   }
}
```

Де змінна host – адреса серверу, котрий буде обробляти і віддавати інформацію від датчиків.

Для реалізації даної схеми було написано простий веб-сервер на мові програмування PHP. Сервер приймає GET-запрос з параметром push_data=true на обновлення інформації з датчика та записує його в файл. Вихідний код серверу приведено нижче:

```
<?php
$file = 'data.txt';
if (!empty($_GET['push_data'])) {
    $tempCelsium = $_GET['c'];
    $tempFarengeit = $_GET['f'];
    $humidity = $_GET['h'];
    $arrayData = [
        'celsium' => $tempCelsium,
        'farengeit' => $tempFarengeit,
        'humidity' => $humidity,
    ];
    file_put_contents($file, json_encode($arrayData,
JSON_UNESCAPED_UNICODE));
} else {
    $currentData = json_decode(file_get_contents($file), true);
    if ($currentData) {
        $tempCelsium = $currentData['celsium'];
        $tempFarengeit = $currentData['farengeit'];
        $humidity = $currentData['humidity'];
        $page = "<html><head></head><body><h3>Temperature in Celsius:
 $tempCelsium
*C</h3><h3>Temperature in Fahrenheit:
$tempFarengeit*F</h3><h3>Humidity:
 $humidity%</h3><h3>
```

```
</h3></body></html>";
    }
    echo $page;
}
?>
```

Для відображення даних від датчиків на сторінці було створенно iframe-вікно, яке дає можливість відображення інформації з довільного домену та відмінного від хосту самого сайту ІТС. На рисинку 3.16 зображено успішне зчитування інформації з датчиків та відображення їх на сайті.

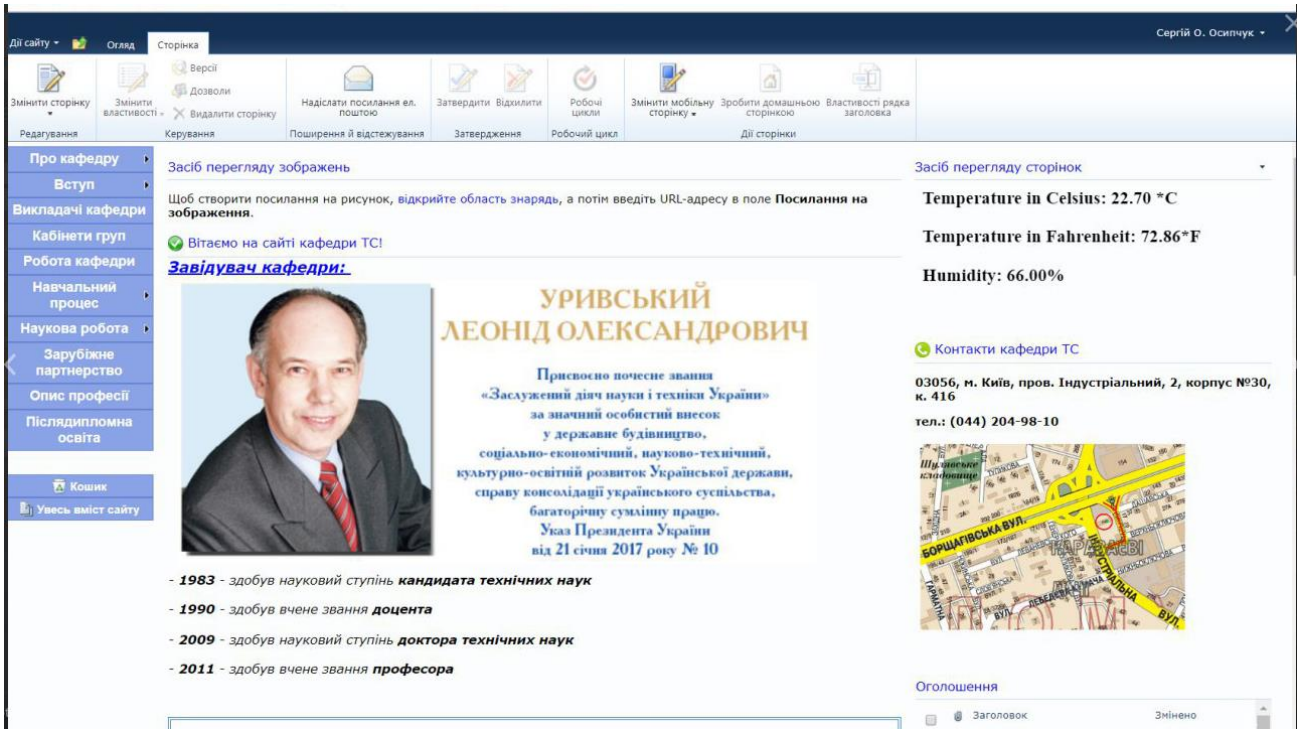

Рисунок 3.16 Відображення температури та вологості з датчика DHT11 на сторінці кафедри ТС

3.4 Висновки з розділу 3

ETL процеси відбуваються на кількох рівнях - периферійних пристроях, шлюзів і серверного рівня. Поки немає єдиного підходу, що повинно робитися на кожному рівні, але ідеологія така - все що можна обробити має бути оброблено на якомога нижчому рівні.

Основним, універсальним засобом передачі інформації є бездротовий інтернет. Але фактичні протоколи відрізняються на різних рівнях. Логічний рівень зв'язку - протокол LvM2M.

Серверний рівень більшою мірою Cloud-орієнтований. Більшість рішень на сьогоднішній день використовують AWS.

Як пристрій відображення фактичної інформації використовується мобільний додаток, а аналітична інформації представляється WEB додатками.

Також в даному розділі представлено результати по практичній реалізації шаблону взаємодії IoT «запит-відповідь».

Отже, було успішно створенно та розгорнено мережу з платою ESP8266 і датчиком DHT11 та серверною частиною на віддаленому хості. Дане рішення дозволяє переглядати данні з датчиків на сайті в режимі реального часу. Також дане рішення може бути легко видозміненне під конкретні датчики і кінцевих клієнтів(сайти, бази даних, мобільні термінали та додатки). Серверна частина також легко модифікується і розширюється в залежності від поставленої завдання.

РОЗДІЛ 4 РОЗРОБКА КОМУНІКАЦІЙНОГО МОДУЛЯ ТА СИСТЕМИ КОНТРОЛЮ ДЛЯ ЗАРЯДНИХ СТАНЦІЙ

4.1 Постановка задачі та актуальність

З розвитком нових енергетичних транспортних засобів у техніці та промисловості, а також фінансовими стимулами уряду та підтримкою відповідної політики нові енергомобілі постійно популяризуються та розвиваються, особливо електромобілі. Все більше людей обирають електромобілі як інструменти для подорожей. Однак зарядні пристрої недосконалі, нестандартні і стандарти зарядки не уніфіковані. Не існує уніфікованого протоколу зв’язку між різними виробниками зарядних станцій, що ускладнює популяризацію іновацій. Ці фактори обмежують розвиток нових енергетичних транспортних засобів, особливо електромобілів. Поява стандартизації OCPP 1.6 забезпечує практичне та ефективне рішення для інтеграції та глобалізації протоколу зарядки.

Розробка універсального комунікаційного модуля для зарядних станцій електромобілів, який буде працювати згідно протоколу OCPP з метою телеметричного та інших видів контролю і управління станцією, є дуже важливою частиною по розвитку та впровадженню електромобілів.

4.2 Огляд протоколу OCPP

4.2.1 Вступ

Після фінансової кризи провідні країни світу розглядають розвиток електромобільної промисловості як свою національну стратегію, а розвиток нових галузей стає важливим способом полегшення енергетичного кризису [9]. Китай перевершує Сполучені Штати і стає найбільшою в світі країною з виробництва електромобілів у виробництві та продажах за останні роки. Як чистий енергетичний засіб транспорту, електромобілі стали все популярнішими в усіх сферах завдяки своїм низьким цінам на пальне та безпеці експлуатації. Однак реальним обмеженням для електромобіля є побудова зарядного пристрою та

зручність зарядки. Стандартизація та уніфікація протоколу зарядного пристрою, взаємозв'язок між обладнаннями та передача даних між виробниками зарядних станцій є основними проблемами. Протокол OCPP 1.6 (відкритий протокол зарядки), запропонований Альянсом відкритого зарядження (OCA), застосовувався до більш ніж 40 000 зарядних установ у 49 країнах, він став глобальним стандартом. Далі розглядається даний протокол та реалізація передачі повідомленнь згідно протоколу.

4.2.2 OCPP 1.6

Що таке OCPP (Open Charge Point Protocol).

Відкритий зарядний альянс (OCA) - міжнародна партнерська організація, яка включає провідних та ділових лідерів в громадських та приватних областях електричної інфраструктури транспортних засобів. Метою Альянсу відкритого зарядження є сприяння розробці та застосуванню протоколу зарядки електроустановок для зарядки електромобілів шляхом співпраці, навчання, тестування та сертифікації, а також сприяння галузевій стандартизації відповідних угод. Протокол відкритої зарядки (OCPP) - це стандарт відкритого зв'язку, запроваджений Альянсом відкритої зарядки. В основному використовується для вирішення труднощів зв'язку між приватними зарядними мережами. OCPP підтримує зв’язок між пунктом зарядки та центральною системою управління провайдера. OCPP використовується в більш ніж 40 000 зарядних станцій у 49 країнах.

OCPP 1.6 заснований на OCPP 1.5. OCPP 1.5 широко використовується у світі з 2012 року, і багато постачальників застосовують стандарт OCPP 1.5 у своїх продуктах. Цей досвід застосування додано до OCPP 1.6. Всього 19 компаній внесли свій досвід роботи в OCPP1.6.

Відмінності між OCPP 1. 6 та OCPP 1.5

OCPP 1.6 представляє нові можливості: смарт-зарядку, OCPP за допомогою JSON над WebSockets, краще діагностичні можливості, більш точковий заряд, Статуси і TriggerMessage. OCPP 1.6 базується на OCPP 1.5 з деякими новими можливостями та безліччю текстових удосконалень, уточнень та виправлень для всіх відомих неясностей (issues). Завдяки вдосконаленням та новим функціям, OCPP 1.6 не сумісний із OCPP 1.5.

Смарт-випадки використання зарядних пристроїв

Згідно з OCPP 1.6, систему зарядки можна розділити на дві основні частини - центральну систему та точку заряду. Точка заряду може мати декілька роз'ємів для з'єднання з кількома електричними транспортними засобами. Основна функція протоколу полягає в здійсненні інформаційного зв’язку між Центральною системою і Точкою заряду, щоб стан і відповідні параметри кожної транзакції пункту зарядки були під контролем центральної системи.

Для розумної зарядки може бути багато різних застосувань. Наступні три типові види смарт-зарядки будуть використані для ілюстрації можливої поведінки інтелектуальної зарядки [10].

Топологія випадку балансування навантаження

Випадок використання балансування навантаження стосується внутрішнього врівноваження навантаження в точці заряду, зарядний пункт керує графіком зарядки для кожного з'єднувача. Точка заряду налаштована з фіксованою межею, наприклад, максимальним струмом підключення до мережі.

Необов'язкове поле графіку зарядки minChargingRate може використовуватися зарядною точкою для оптимізації розподілу потужності між роз'ємами. Параметр інформує пункт заряду, що зарядка нижче minChargingRate є неефективною, що дає можливість вибору іншої стратегії балансування [10].

Топологія показана на рисунку 4.1.

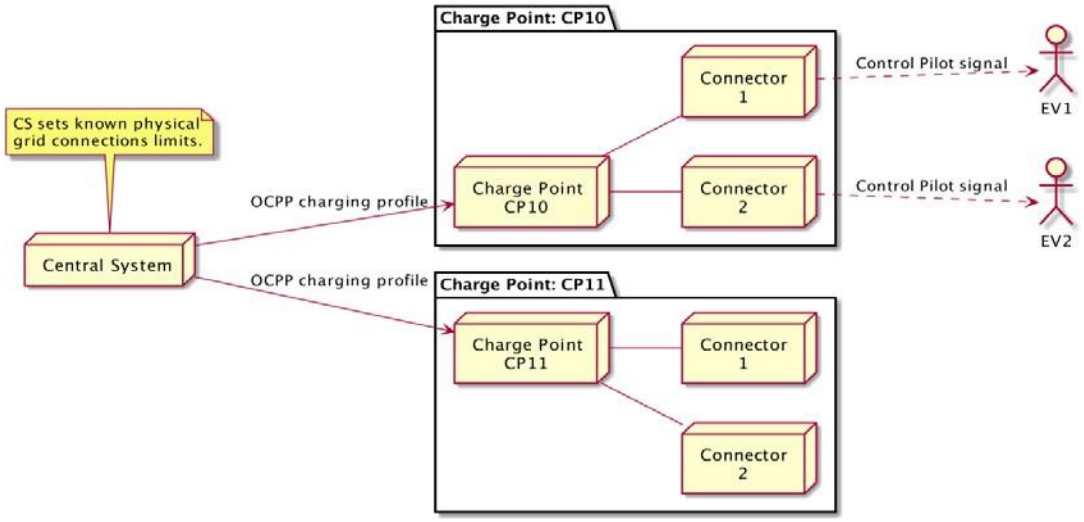

Рисунок 4.1 Балансування навантаження топології інтелектуального заряджання

Центральна інтелектуальна зарядка

При центральній інтелектуальній зарядці обмеження в графіку зарядки за транзакцією визначаються Центральною системою. Центральна система використовує ці графіки, щоб залишатися в межах, встановлених будь-якою зовнішньою системою. Центральна система безпосередньо контролює обмеження на роз'ємах пунктів заряду [10].

Центральна розумна зарядка передбачає, що обмеження заряду контролюються центральною системою. Центральна система отримує прогноз потужності від оператора мережі (DSO) або іншого джерела в тій чи іншій формі і розраховує графіки зарядки для деяких або всіх операцій з оплати, деталі яких не виходять за межі цієї специфікації.

Топологія показана на рисунку 4.2.

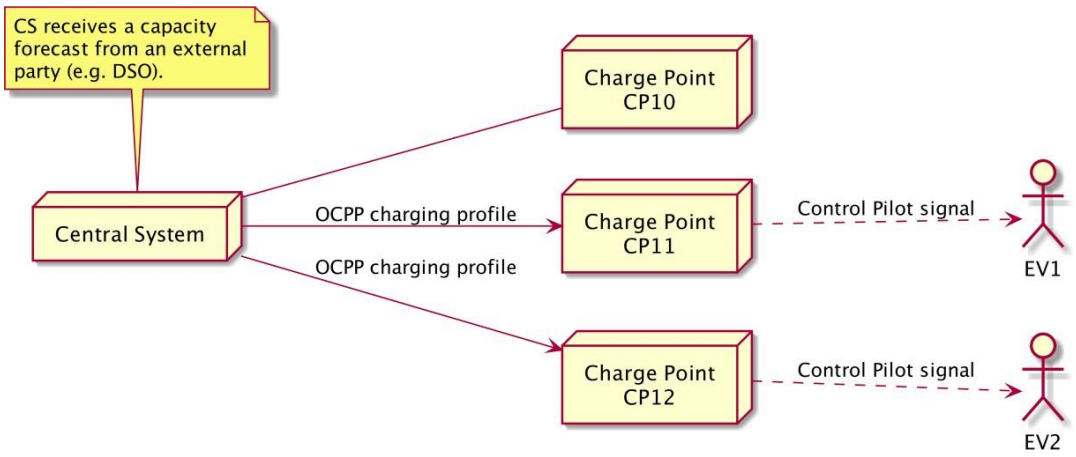

Рисунок 4.2 Центральна топологія інтелектуального заряджання.

Локальна інтелектуальна зарядка

Випадок використання локальної інтелектуальної зарядки описує випадок використання, у якому пункти зарядки з увімкненою смарт-зарядкою мають обмеження зарядки, які контролюються локальним контролером, а не центральною системою. Випадок використання локальної розумної зарядки - це обмеження кількості енергії, яку може використовувати група зарядних точок, до певного максимуму. Типовим способом використання буде кількість точок заряду в паркувальному гаражі, де рейтинг підключення до мережі менший за суму оцінок пунктів зарядки. Іншою програмою може бути те, що локальний контролер отримує інформацію про доступність живлення від DSO або локального вузла розумної сітки.

Локальна розумна зарядка передбачає існування локального контролера для управління групою пунктів нарахування. Локальний контролер - це логічний компонент. Він може бути реалізований або як окремий фізичний компонент, або як частина "головного" пункту зарядки, що контролює ряд інших пунктів зарядки. Місцевий контроль реалізує протокол OCPP і є проксі-сервером для повідомлень OCPP членів групи, і може не мати власних з'єднувачів.

У разі локальної інтелектуальної зарядки місцевий контролер накладає обмеження зарядки на точку заряду. Ці ліміти можуть динамічно змінюватися під час процесу зарядки з метою збереження споживання енергії групи зарядних пунктів у межах групових лімітів. Групові обмеження можуть бути попередньо

налаштовані в локальному контролері або можуть бути налаштовані центральною системою.

Топологія показана на рисунку 4.3.

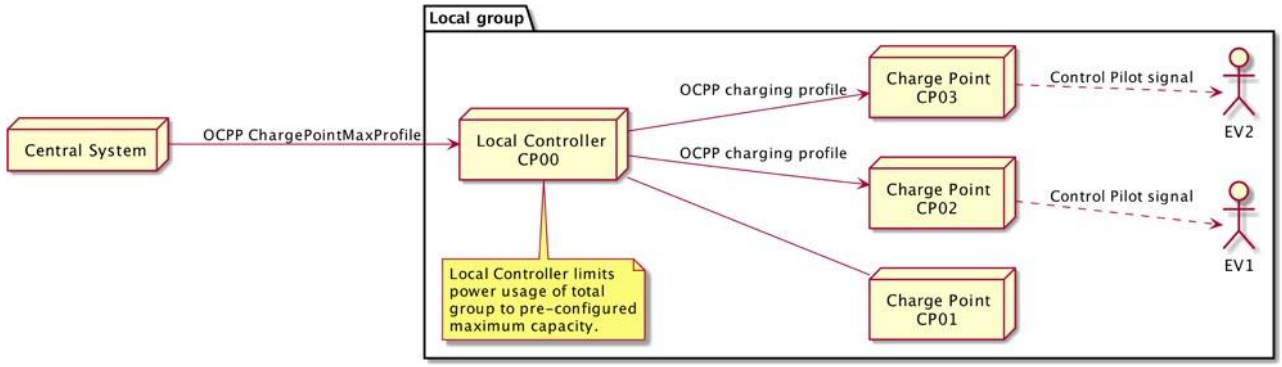

Рисунок 4.3 Локальна топологія інтелектуального заряджання

Вступ до пов'язаних параметрів передачі

У цій частині буде представлена функція передачі та пов'язані з нею параметри, визначені в OCPP 1.6.

У OCPP 1.6 центральній системі та пункту зарядки потрібно передавати та приймати повідомлення один одному, тобто надсилати повідомлення запиту та підтверджувати повідомлення. Оскільки передача повідомлень є взаємною, всі повідомлення можна розділити на два типи, тобто повідомлення, що надсилаються Центральною системою, та повідомлення, що надсилаються пунктом зарядки [11]. Центральна система надсилає повідомлення для отримання поточної інформації про стан або керуючої дії пункту заряду [12]. Повідомлення, надіслане Charge Point, стосується більше інформації про початок та кінець трансакції та серцебиття, значення вибірки та інші дані, пов'язані з бізнесом. Нижче наведено запит Charge Point та запит центральної системи.

Запит зарядного пункту: Heartbeat.req

Точка заряду відправляє серцебиття після настроюваного часового інтервалу, щоб повідомити Центральній системі про те, що точка заряду все ще підключена. Charge Point надішле запит Heartbeat.req, який не містить жодних параметрів. Після отримання цього запиту Центральна система поверне повідомлення про підтвердження Heartbeat.conf (у поточний час), яке вказує, що

Центральна система отримала запит серцебиття та відповіла. Ця відповідь містить параметр з назвою currenttime, який вказує час отримання запиту та підтвердження. Це дані типу DateTime, визначені в протоколі. Схема передачі така, як показана на рисунку 4.

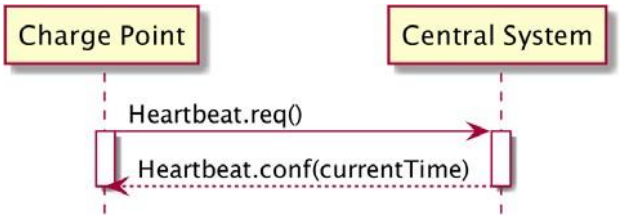

Рисунок 4.4 Діаграма послідовності: серцебиття.

Запит на центральну систему: ReserveNow.req

Центральна система може видати ReserveNow.req до пункту зарядки, щоб резервувати роз'єм для використання певним idTag.

Для запиту бронювання центральна система надсилає PDU ReserveNow.req (блок даних протоколу) до пункту зарядки. Центральна система може вказати роз'єм, який потрібно зарезервувати. Після отримання PDU ReserveNow.req, зарядний пункт відповідає на PDU ReserveNow.conf. Він несе параметр з назвою статус. Необов’язковими значеннями є: Прийнято, Помилка, Зайнято, Відхилено, Недоступно. Діаграма передач наведена на рисунку 4.5.

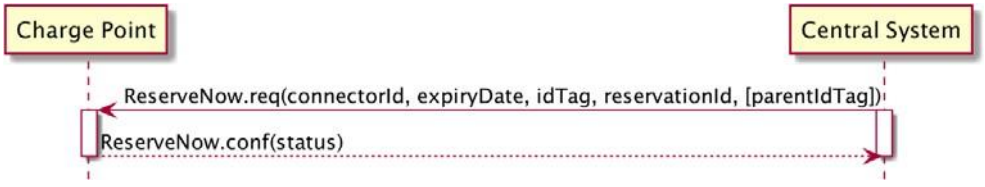

Рисунок 4.5 Діаграма послідовності: Забронювати зараз

4.2.3 Технологія, що використовується для впровадження OCPP 1.6

Через читання та дослідження протоколу OCPP1.6 я створив веб-сервер, щоб центральна система і зарядна точка могли спілкуватися між собою.

Основними технологіями, які використовуються в цьому проекті, є Jsp (динамічна веб-технологія), MySQL, HTML + CSS технологія, технологія JavaScript та Maven (інструмент управління проектами).

4.2.4 Демонстрація доставки повідомлень протоколу

Розгортання проекту

Для здійснення передачі повідомлень між пунктом заряджання та центральною системою був написаний код на стороні клієнта. Процес створення сервера полягає в наступному.

- Спочатку скачайте код проекту з GitHub за наступною адресою: https://github.com/RWTH-i5-IDSG/steve.
- Створіть базу даних з іменем stevedb у локальній базі даних.
- У папці steve-master проведіть компіляцію проекту за допомогою Maven-a.
- Введіть "mvn package", щоб створити цільовий каталог, скомпілювати та протестувати код.
- Після запуску всіх команд виконайте наступну команду у вікні команд DOS для виконання пакету jar: Java -jar target / steve-2.1.0.jar.

Сторінка входу

Ця сторінка в основному реалізує вхід адміністратора, щоб адміністратор міг відслідковувати інформацію та стан пункту зарядки та контролювати центральну систему для надсилання повідомлень до пункту зарядки. Скріншот зображений на рисунку 4.6.

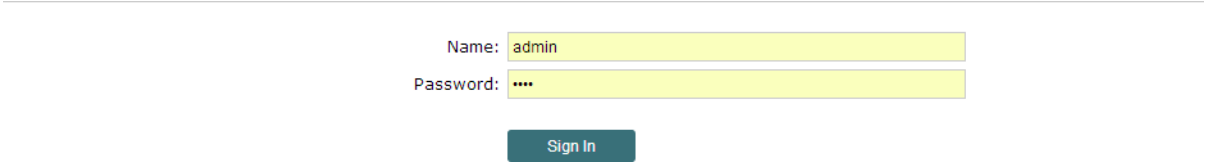

Рисунок 4.6 Сторінка входу

Сторінка реєстрації користувача

Сторінка реєстрації користувачів використовується для реєстрації інформації про цих користувачів, які можуть використовувати пункт зарядки. Після того як інформація про користувачів зареєстрована, Центральна система може активувати послугу зарядки, авторизуючи аутентифікацію після того, як зарядний пункт ініціює запит аутентифікації до Центральної системи, і авторизований користувач може здійснити зарядку.

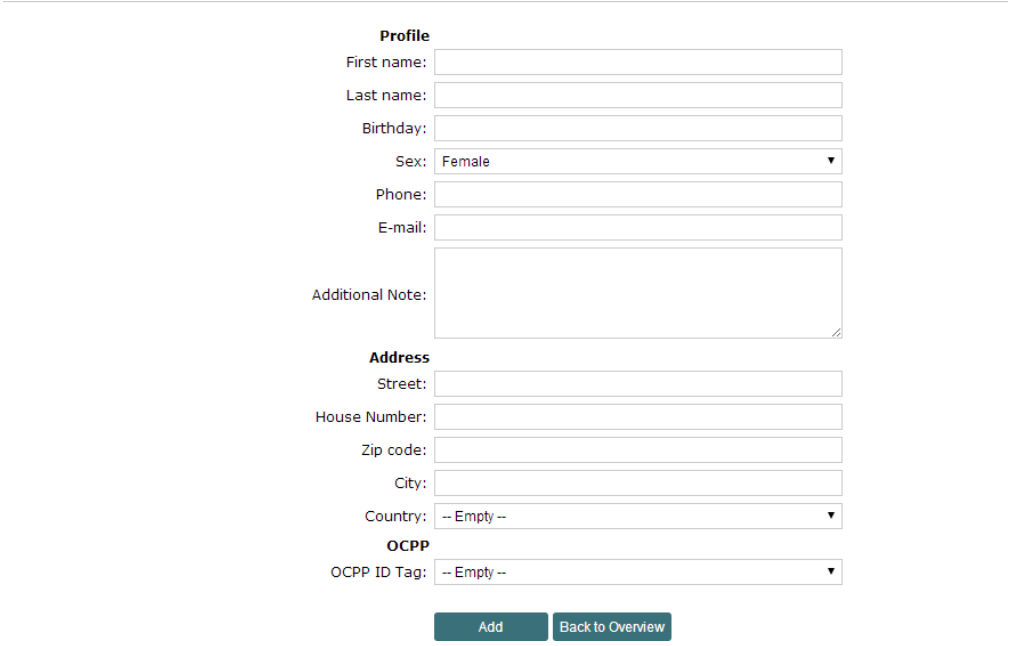

Рисунок 4.7 Сторінка реєстрації користувача

Сторінка реєстрації зарядних пунктів

Ця сторінка завершує реєстрацію пункту зарядки та однозначно ідентифікує її ідентифікатор, щоб Центральна система могла точно контролювати пункт зарядки. Ця сторінка також містить таку інформацію, як місце розташування Charge Point та ін. Скріншот зображений на рисунку 4.8.

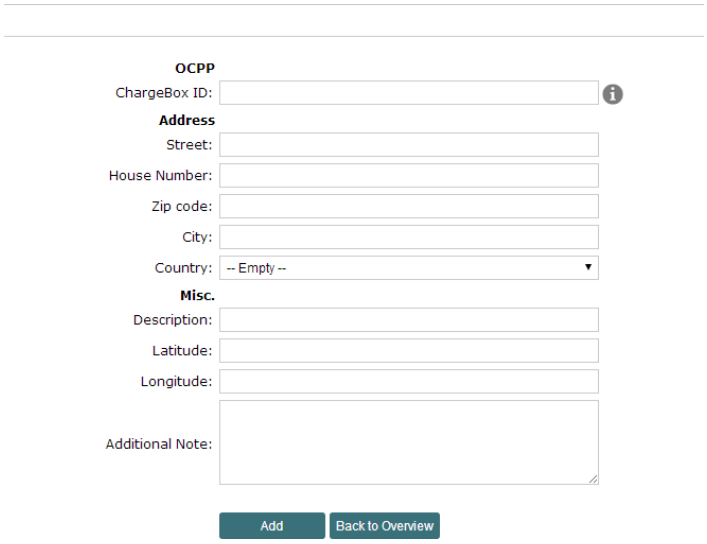

Рисунок 4.8 Сторінка реєстрації зарядних пунктів

Сторінка огляду зарядних пунктів

На сторінці огляду пункту зарядки перераховані відомості про зарядку та час останнього запиту серцебиття.

Рисунок 4.9 Сторінка огляду пункту зарядки

Сторінка огляду інформації про стан зарядки

На цій сторінці перераховано кількість користувачів, кількість зарядних пунктів, діяльність зарезервованих та кількість поточних ЗС, інтерфейси зарядки, серцебиття та інша інформація. Ця сторінка полегшує загальний контроль адміністратора.

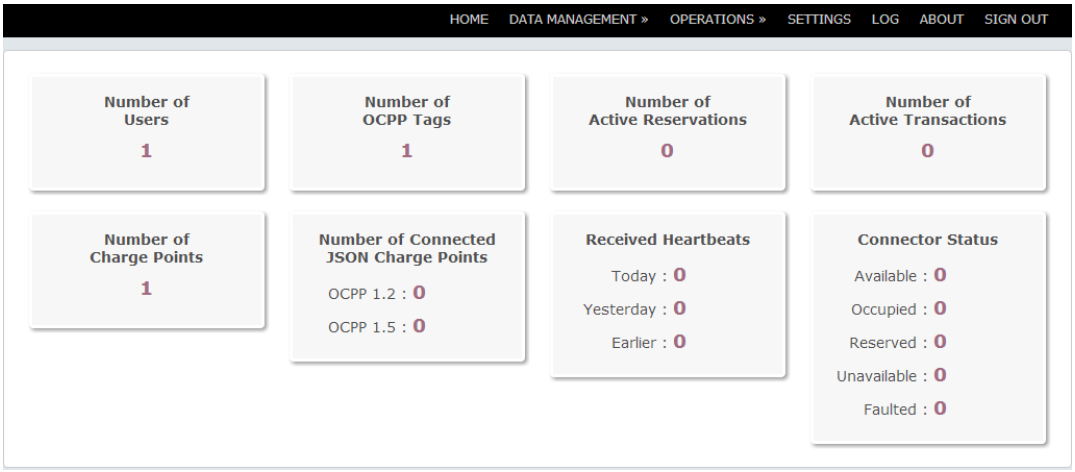

Рисунок 4.10 Сторінка огляду інформації про стан зарядки

Надсилання запиту серцебиття

На малюнку нижче показано клієнт SoapUI, що надсилає серцебиття на сервер, і сервер, що приймає серцебиття. Порт, на якому сервер отримує повідомлення: http://localhost:8080/steve/services/CentralSystemServiceOCPP15.

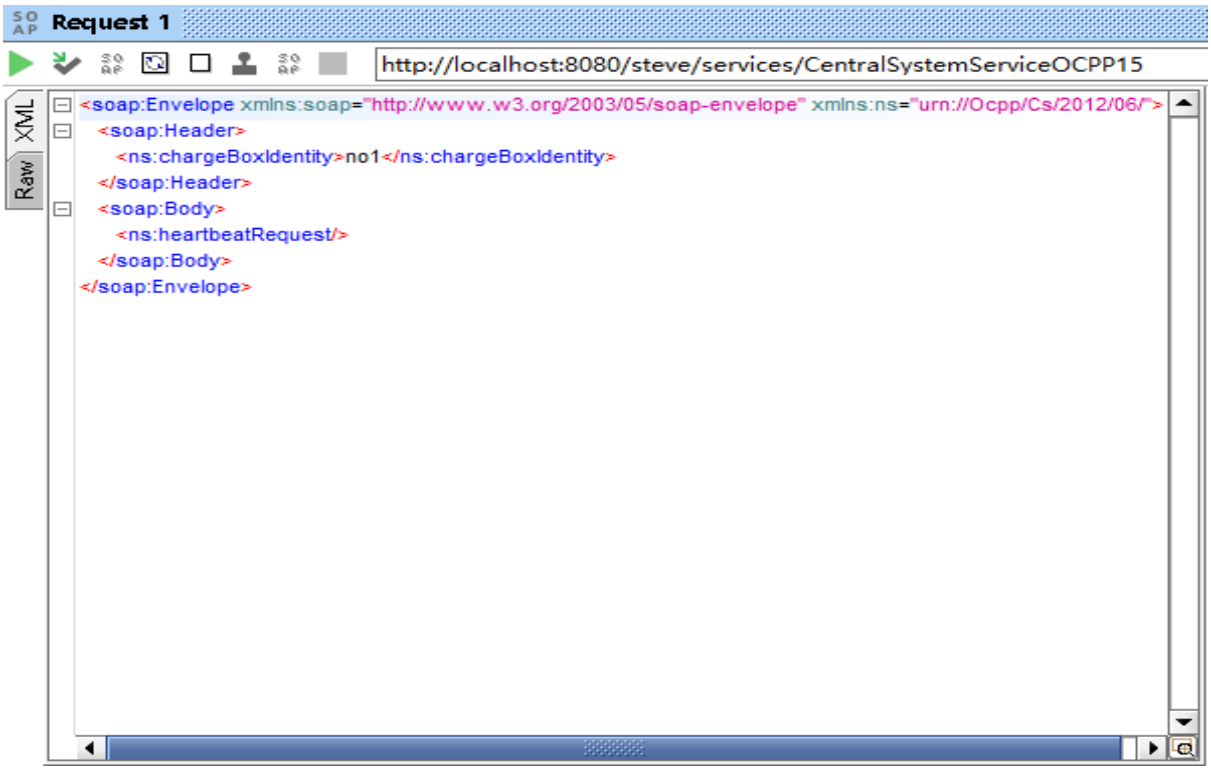

Рисунок 4.11 Надсилання запиту на серцебиття

Після успішного надсилання повідомлення центральна система отримає повідомлення, відображене на сторінці статусу пункту зарядки, як показано на рисунку 4.12.

Рисунок 4.12 Отримання запиту на серцебиття

Надсилання ReserveNow Request

Сторінка повідомлення, що надсилається, відображається так:

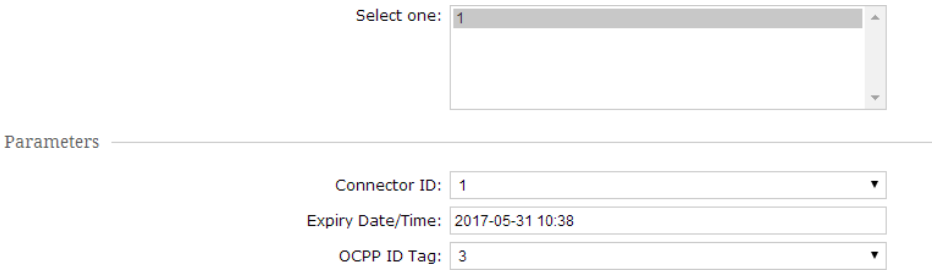

Рисунок 4.13 Надсилання запиту ReserveNow

Центральна система хоче надіслати запит ReserveNow до роз'єму №1 пункту заряду №1. Якщо запит ReserveNow не обробляється, коли запит перевищує обмеження, запит ReserveNow закінчується.

Коли повідомлення надіслано, його статус стає в очікуванні, очікуючи на відповідь. Скріншот виглядає наступним чином.

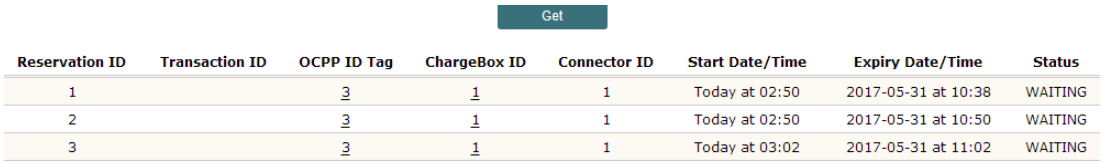

Рисунок 4.14 Статус запиту ReserveNow

Як тільки запит ReserveNow приймається, клієнт повертає поле статусу відповідно до статусу з'єднувача. Рисунок 4.15 вказує на те, що бронювання було успішним, і з'єднання було зарезервоване.

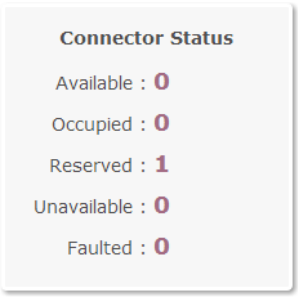

Рисунок 4.15 Запит ReserveNow успішно надісланий

4.3 Архітектура системи моніторингу та управління зарядних станцій

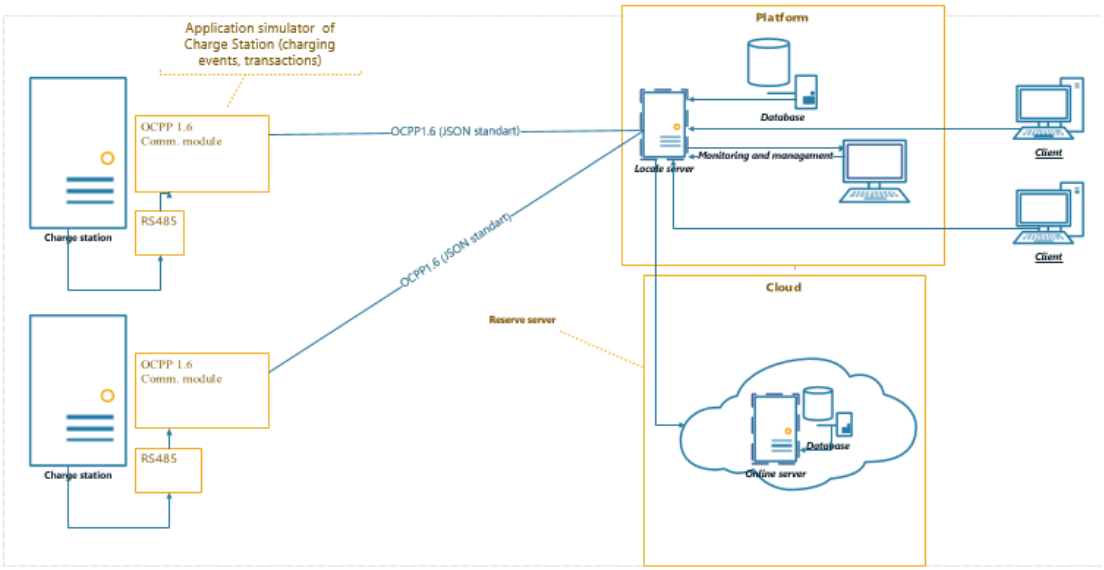

Рисунок 4.16 Архітектура системи моніторингу та управління зарядних станцій

Дана схема (рисунок 4.16) системи управління зарядними станціями складається з серверної частини, пула комунікаційних модулів зарядних станцій, та клієнтів платформи.

Серверна частина складається з серверу, написаного мовою Java та бази даних MySQL. Сервер підтримує створення сокет-сесій та протоколи OCPP1.6 для обміну даних від ЗС. Для клієнтів є можливість використання веб-інтерфейсу для

перегляду підключених зарядних станцій(рисунок 4.17), списку транзакцій (рисунок 4.18) тощо.

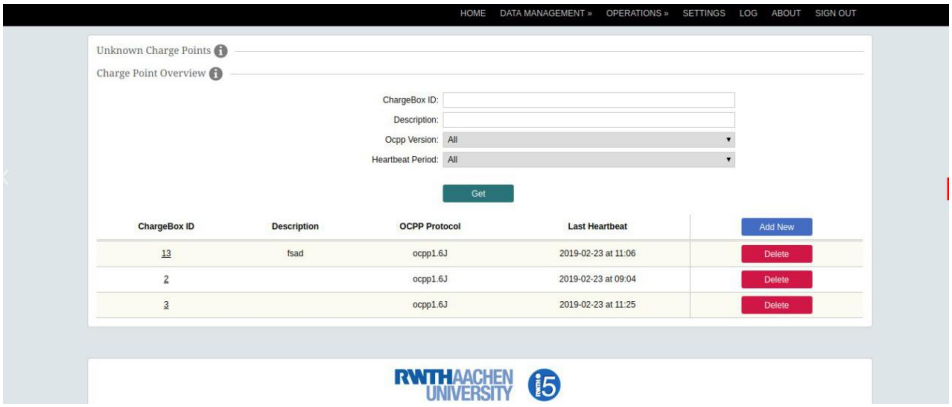

Рисунок 4.17 Сторінка перегляду підключених зарядних станцій

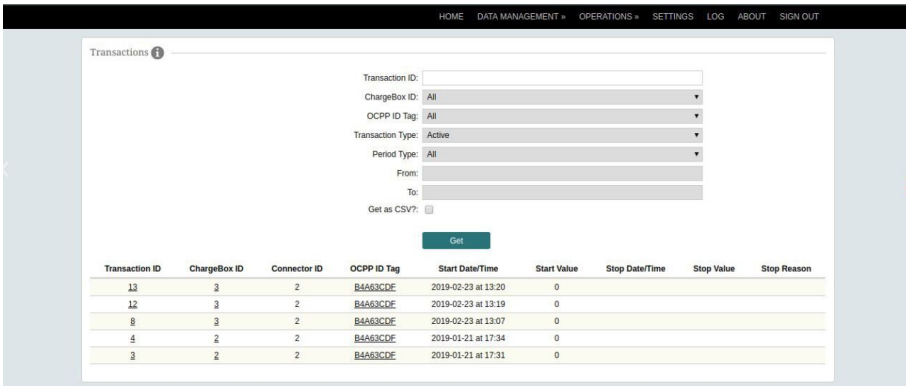

Рисунок 4.18 Сторінка перегляду списку транзакцій

Пул комунікаційних модулів ЗС представлений набором програмних симуляторів, які імітують повний функціонал зарядної станції відповідно до протоколу OCPP 1.6, такі як авторизація ЗС, початок/зупинка транзакцій (процесу заряду), опитування зарядної станції зі сторони сервера, heartbeat.

Передача даних від ЗП відбувається з допомогою протоколу HTTP/2 над протоколом TCP/IP, а також протокол WebSocket - протокол, що призначений для обміну інформацією між клієнтом та веб-сервером в режимі реального часу. Він забезпечує двонаправлений повнодуплексний канал зв'язку через один TCP-сокет. В якості формату обміну даних було обрано JSON (англ. *JavaScript Object Notation)*

Для моніторингу пула ЗС можна використовувати веб-інтерфейс. В ході експериментального моделювання було створено та підключенно декілька симуляторів ЗС до сервера, та відпрацьовано процеси та функції згідно протоколу OCPP 1.6.

4.4 Висновки з розділу 4

Все більше людей обирають електромобілі як інструменти для подорожей. Однак зарядні пристрої недосконалі, нестандартні і стандарти зарядки не уніфіковані. Не існує уніфікованого протоколу зв’язку між різними виробниками зарядних станцій, що ускладнює популяризацію іновацій. Ці фактори обмежують розвиток нових енергетичних транспортних засобів, особливо електромобілів. Поява стандарту OCPP 1.6 забезпечує практичне та ефективне рішення для інтеграції та глобалізації протоколу зарядки. Даний стандарт можливо стане основним стандартом при розробці мереж зарядних станцій.

В даному розділі було розглянуто протокол OCPP та реалізація передачі повідомленнь згідно стандарту, оглянуто основні можливості та функції даного протоколу. Також було розроблена архітектура системи моніторингу та управління зарядних станцій, створена імітаційна модель комунікаційного модуля з повною підтримкою протоколу OCPP 1.6. В ході експериментального моделювання було створено та підключенно декілька симуляторів ЗС до сервера, та відпрацьовано процеси та функції згідно протоколу OCPP 1.6.

ВИСНОВКИ

З кожним роком вимоги до пропускної здатності та якості безпроводових телекомунікаційних мереж стають більш вимогливими, тому при створені сучасних беспроводових систем пріоритетними задачами мають стати:

– розробка нових технологій для гнучкого використання спектру та мобільного широко-смугового доступу;

– інтеграція технологій радіозв'язку з волоконно-оптичними мережами, для об'єднання мобільних і бездротових мереж в комплексні системи зв'язку з метою забезпечення високошвидкісного бездротового доступу в усіх галузях діяльності людини;

– мережі мають бути готовими до забезпечення обміну нетипової для них інформації в нових областях застосування;

– забезпечення надійного захисту інформації;

– створення нової системи управління інфокомунікаційними мережами.

Впровадження Інтернету речей – це важлива частина при створенні сучасних інфокомунікаційних систем і мереж. Впровадження інтернету речей відбуваються поки не в глобальних масштабах, а всередині компаній, корпоративних мережах тощо. Технологія розумних речей здатна підвищити продуктивність праці в першу чергу в виробничому сегменті, логістичному бізнесі, транспортних і енергетичних компаніях.

З чого починати будувати архітектурне рішення IoT? Немає єдиного підходу до відповіді на це питання. Далі приведено деякі рекомендації:

- Визначити модель даних, яку ми можемо отримати від шлюзу, тобто передану в серверну частину рішення.
- Перевірити які саме пристрої можуть зібрати дані і як їх треба обробити для приведення до моделі переданої шлюзом.
- Перевірити вимоги до пристроїв - відстані, обсяг інформації, енергоспоживання та ін.

- Вибрати відповідний обчислювальний пристрій, його розташування по відношенню до датчиків, протокол їх роботи.

Вирішити архітектуру хмарної частини, якщо вона присутня в архітектурі рішення, включаючи:

- Безпека
- розподіл навантаження
- Асинхронність передачі даних усередині Cloud-мережі
- Елементи зберігання, форму і життєвий цикл даних
- Побудувати граф передачі інформації по системі
- Побудувати аналітичні моделі, AI/ML компонент
- Розробити типи і зміст повідомлень
- Налаштувати резервування і авто масштабованість сервісів
- Оцінити вартість і провести оптимізацію
- Дизайн UI / UX для мобільних клієнтів
- Побудувати зворотний зв'язок передачі даних в периферійний пристрій

Також в даній роботі представлено результати по практичній реалізації шаблону взаємодії IoT «запит-відповідь». Отже, було успішно створенно та розгорнено мережу з платою ESP8266 і датчиком DHT11 та серверною частиною на віддаленому хості. Дане рішення дозволяє переглядати данні з датчиків на сайті в режимі реального часу. Також дане рішення може бути легко видозміненне під конкретні датчики і кінцевих клієнтів(сайти, бази даних, мобільні термінали та додатки). Серверна частина також легко модифікується і розширюється в залежності від поставленої завдання.

В 4 розділі було розроблена архітектура системи моніторингу та управління зарядних станцій, створена імітаційна модель комунікаційного модуля з повною підтримкою протоколу OCPP 1.6. В ході експериментального моделювання було створено та підключенно декілька симуляторів ЗС до сервера, та відпрацьовано процеси та функції згідно протоколу OCPP 1.6.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Макаренко А.Ю. Бездротові технології передачі даних Wi-Fi, Bluetooth та ZigBee. / А.Ю. Макаренко, А.О. Парфенова, С.Б. Могильний // Вісник НТУУ «КПІ». Серія Радіотехніка. Радіоапаратобудування. – 2010. – № 41. – с. 171-181.

[2] IEEE Std 802.11a-1999 (Reaff 2003), Supplement to IEEE Standard for Information technology— Telecommunications and information exchange between systems—Local and metropolitan area networks— Specific requirements.

[3] IEEE Std 802.11b-1999, Supplement to IEEE Standard for Information technology— Telecommunications and information exchange between systems—Local and metropolitan area networks— Specific requirements.

[4] IEEE Std 802.11g-2004, IEEE Standard for Information technology— Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements— Specifications.

[5] IEEE Std 802.11i-2004, IEEE Standard for Information technology— Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements.

[6] IEEE Std 802.11n-2009, IEEE Standard for Information technology— Telecommunications and information exchange between systems—Local and metropolitan area networks.

[7] IEEE Std 802.15.1-2002, IEEE Standard for Information technology— Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements.

[8] IoT архитектура / https://habr.com/ru/post/455377/

[9] Cristina Alcaraz, Javier Lopez, Stephen Wolthusen (2017) OCPP Protocol: Security Threats and Challenges. IEEE Transactions on Smart Grid, 2, 1-1.

[10] Open Charge Allianc (2015) Open Charge Point Protocol 1.6.

[11] Á.Rodríguez-Serrano, A.Torralba, E.Rodríguez-Valencia, J.Tarifa-Galisteo (2013) A communication system from EV to EV Service Provider based on OCPP over a wireless network. IEEE, 10, 5434-5438.

[12] Jens Schmutzler, Claus Amtrup Andersen, Christian Wietfeld (2013) Evaluation of OCPP and IEC 61850 for Smart Charging Electric Vehicles. Electric Vehicle Symposium and Exhibition, 10, 1-12.

[13] T. Parker, D. Naberezhnykh (2013) Charging point strategies for electric commercial vehicles. Hybrid and Electric Vehicles Conference, 10, 1-4.